



**TeleFlow<sup>®</sup>**

**Printed Manual Format  
for TeleFlow Help**

Copyright ©2008 enGenic

**engen<sup>i</sup>c**

# Table of Contents

Foreword	0
<b>Part I System Requirements</b>	<b>2</b>
<b>Part II TeleFlow Installation</b>	<b>4</b>
<b>Part III TeleFlow Overview</b>	<b>8</b>
<b>Part IV Licensing Requirements</b>	<b>11</b>
<b>Part V 1- Develop Applications with TeleFlow Designer</b>	<b>13</b>
<b>1 What is the TeleFlow Designer?</b>	<b>13</b>
<b>2 Flowcharts</b>	<b>14</b>
<b>3 Getting Started</b>	<b>15</b>
TeleFlow Basics	15
Creating A New Application	16
Opening an Existing Application	18
Adding Flowcharts to an Application	18
Adding a New Flowchart	18
Adding an Existing Flowchart	19
Setting the First Flowchart in an Application	20
Adding Step Icons to Flowcharts	20
<b>4 Getting Around in TeleFlow Designer</b>	<b>21</b>
Flowchart Screen Menus	22
Quick Map	27
Steps Toolbox Window	27
Application Environment Settings	28
Search Steps for Text	30
Hot Menus	33
Step Options Hot Menu	33
Selected Area Hot Menu	34
Tablet Options Hot Menu	34
<b>5 Hints, Tips, and Tricks</b>	<b>35</b>
Isolating a Repetitive Series of Icons into a Separate Flowchart	35
Managing Repetitive Voice Scripts	36
Setting Icons Up So The Flow is Easy to Follow	37
Using Labels	37
Using the Connect Point Step	38
<b>6 Steps Toolbox</b>	<b>38</b>
TeleFlow Steps	40
START and FINISH steps	43
Starting Point	43
START step	43
FINISH step	44
Action Steps	45

Busy .....	45
Connector Icon .....	45
Failure Action .....	46
Invalid Action .....	46
Telephone Buttons .....	47
Timeout Action .....	48
Unrecognized Voice Action .....	48
True / False or Yes / No Action Steps .....	49
Phone Steps .....	49
Block Port .....	50
Conference .....	50
Open Conference .....	50
Join Conference .....	51
Leave Conference .....	53
Close Conference .....	53
Disable Reorder .....	54
Enable Reorder .....	54
Fax Steps .....	55
Add to Fax .....	55
Fax Clear .....	55
Fax Receive .....	56
Fax Send .....	57
Flash Hook .....	57
Global Event .....	58
Hang Up .....	59
Pick Up .....	60
Place Call .....	60
Register SIP URI .....	62
Send TT .....	63
Set Media for VoIP .....	63
Set Reorder Tone .....	64
Switch Call .....	66
Switch Disconnect .....	66
Transfer Call .....	67
Unblock Port .....	68
Wait for Call .....	68
Input Steps .....	69
Clear TT .....	70
Get Letter .....	70
Get TT String .....	71
Menu .....	72
Pending TT .....	74
Audio Steps (former Voice tab) .....	75
Beep .....	75
Play .....	76
Record .....	77
Say .....	78
Set Play Variable .....	80
Set Volume .....	80
Speak Text .....	81
Send TT .....	82
Data/XML Steps .....	82
DB Connect .....	83
DB Disconnect .....	84

DB Lock .....	84
SQL Begin Transaction.....	85
SQL End .....	86
SQL End Transaction.....	86
SQL Fetch .....	87
SQL Statement .....	88
XML End .....	89
XML Fetch .....	90
XML Load .....	92
XML Write .....	93
Documentation Steps (Doc. tab).....	94
Connect Point .....	94
Notes .....	95
Save Log .....	95
Step Comments.....	96
Write Text .....	96
Programming Steps (Prog. tab).....	97
Case .....	97
Compare .....	98
Copy File .....	99
Create Folder .....	100
Date Info .....	100
Decrypt .....	101
Delete File .....	102
DLL Function .....	102
Encrypt .....	104
Fatal Stop .....	106
Financial Calc .....	106
Global Event .....	107
Go To Label .....	108
Hash .....	108
Label .....	109
Load File Step.....	109
Write File Step.....	110
Math .....	111
Parse .....	112
Random Number.....	113
Read Registry .....	114
Reload Off .....	114
Reload On .....	115
Run Basic Script.....	115
BAND   BOR   BXOR   BNOT   ^.....	116
AND   OR   XOR   NOT .....	117
CHAR .....	117
CONCAT .....	120
DO   WHILE... LOOP [WHILE   UNTIL].....	120
EXIT .....	122
FOR ... NEXT [STEP] .....	122
GOSUB ... RETURN.....	123
GOTO .....	124
IF...THEN...[ELSEIF]...[, ELSE]..END.....	125
LEFT .....	126
LEN .....	126
LOWERCASE.....	127

MOD .....	127
POS .....	127
PRINT .....	128
REM   // .....	128
RIGHT .....	128
ROUND .....	129
SELECT CASE .....	129
Setting TeleFlow variables in Basic Script .....	131
Standard math operators: + / - * = .....	131
SUBSTR .....	131
TRIM .....	132
UPPERCASE .....	132
Run Flowchart .....	132
Run Program .....	133
Run TAP .....	133
Save Log .....	134
Scan Folder .....	135
Set Global Variable .....	136
Set Language .....	136
Set Local Variable .....	137
Set Play Variable .....	138
Wait .....	138
Write Text .....	139
Advanced Steps (Adv. tab) .....	139
Set HW Parameter .....	139
SR .....	140
Menu .....	141
Speak Text .....	143
SR Acquire .....	144
SR Barge-In Off .....	145
SR Barge-In On .....	145
SR Clear Voice Buffer .....	146
SR Flag Result .....	146
SR Get Digits .....	147
SR Get String .....	148
SR Get Yes/No .....	150
SR Release .....	151
SR VoicePass Auth .....	151
SR VoicePass Reg .....	152
Web Steps .....	154
HTTP .....	154
Send Email .....	155
TCP Connect .....	156
TCP Disconnect .....	157
TCP Receive .....	157
TCP Send .....	158
UDP Bind .....	159
UDP Close .....	160
UDP Receive .....	160
UDP Send .....	161
WebStepsTOC .....	162
<b>7 Variables .....</b>	<b>162</b>
<b>Variable Basics and Scope .....</b>	<b>162</b>
<b>Variable Data Types .....</b>	<b>164</b>

<b>Arrays</b> .....	<b>166</b>
<b>System Variables</b> .....	<b>167</b>
Application Parameters @P1 ... @P9.....	168
CALL_ELAPSE.....	169
CALL_END.....	169
CALL_START.....	169
CALLOUT_REASON.....	169
CALLOUT_RESULT.....	169
CHAR_ or Character variables.....	170
DATE .....	171
DATE_SEPARATOR.....	171
DATE_UTC.....	171
DATETIME.....	172
DATETIME_UTC.....	172
FATAL .....	172
FORMFEED.....	172
HANGUP.....	172
HOOK_STATUS.....	173
LINEFEED.....	173
P1 to P9.....	173
PORT .....	173
PROMPT.....	174
RECORD_RESULT.....	174
SPACE .....	174
SPELLTONES.....	174
SR_END_OF_VOICE_TIMEOUT.....	174
SYS_APP_STATUS.....	175
SYS_CALLREF.....	175
SYS_DATA.....	176
SYSDIR.....	176
SYSDRIVE.....	176
TIME .....	176
TIME_UTC.....	176
TIMEOUT.....	177
TRUNK_TYPE.....	177
UNIX_TIME.....	177
UTC_DATE.....	178
UTC_TIME.....	178
VOLUME.....	178
WEEKDAY.....	179
<b>8 Voice Script Manager</b> .....	<b>179</b>
<b>What is the Voice Script Manager?</b> .....	<b>180</b>
<b>Getting Around in Voice Scripts Manager</b> .....	<b>181</b>
Edit Menu.....	181
File Menu.....	181
Help Menu.....	181
Tools Menu.....	182
Windows Menu.....	182
<b>Adding Voice Scripts to an Application</b> .....	<b>182</b>
<b>Recording Voice Files with the Script Manager</b> .....	<b>182</b>
<b>Using Existing or Pre-Recorded Voice Files</b> .....	<b>183</b>
<b>Editing Voice Scripts</b> .....	<b>183</b>
<b>9 Voice Prompts</b> .....	<b>183</b>

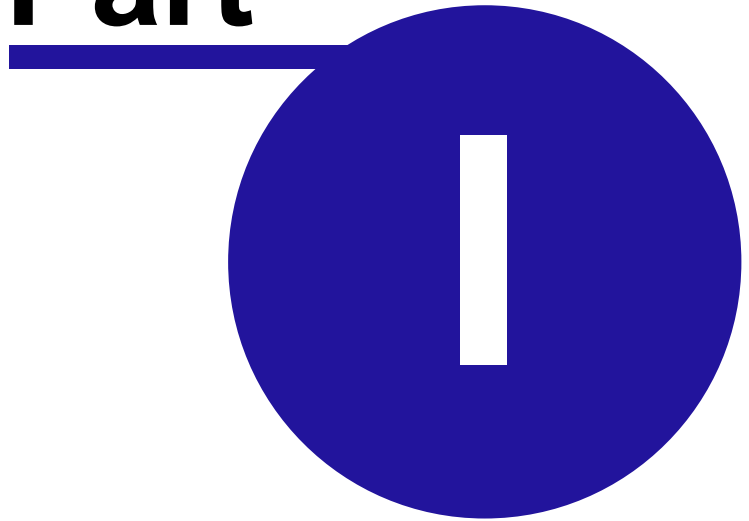
English Standard Word Set .....	183
English to French Standard Word Set .....	188
<b>Part VI 2 - Test Applications with TeleFlow Simulator</b>	<b>195</b>
<b>Part VII 3 - Configure TeleFlow Server</b>	<b>198</b>
1 Development or Production Options .....	198
2 TeleFlow Line List .....	198
3 TeleFlow Config .....	200
4 TeleFlow Register .....	201
<b>Part VIII 4 - Deploy/Run Applications on TeleFlow Server</b>	<b>204</b>
1 Running TeleFlow Server as a Service .....	204
2 Running TeleFlow Server in Console Mode .....	206
3 Stopping TeleFlow Server .....	207
<b>Part IX 5 - Monitor and Maintain TeleFlow Applications</b>	<b>209</b>
1 TeleFlow Monitor .....	209
TeleFlow Monitor Menus .....	211
2 SystemHealth .....	213
Configuration of SystemHealth .....	216
3 Safe shutdown for TeleFlow Server .....	217
<b>Part X Reports and statistics</b>	<b>219</b>
1 How to use TeleFlow Statistics Reporter .....	219
2 TeleFlow SQL Reporter .....	224
3 How to create an SQLReporter file .....	225
<b>Part XI TeleFlow FAQ's</b>	<b>228</b>
1 TeleFlow Designer .....	235
2 Dialogic .....	239
3 TFSimulator .....	239
<b>Part XII Glossary</b>	<b>242</b>
<b>Part XIII Appendix 1 - Speech Recognition</b>	<b>246</b>
1 Planning and Development .....	246
2 Tuning Speech Recognition with TeleFlow SpeechTrainer .....	255
Starting with SpeechTrainer .....	255
How to get SR information .....	258
Recognition Flagged Results file .....	259
Recognition Results file .....	260

Summary File .....	264
<b>Part XIV Appendix 2 - Third Party Links and Requirements</b>	<b>266</b>
1 Lumenvox Speech Recognition .....	266
2 Cool Edit .....	266
3 Microsoft Product Support .....	266
4 Nuance Speech Recognition .....	267
<b>Part XV Appendix 3 - Misc</b>	<b>269</b>
1 Database and ODBC Issues .....	269
ODBC Setup .....	269
Setting up ODBC Data Sources.....	269
SQL .....	269
SQL Notes.....	270
SQL User's Guide.....	271
2 Hardware Installation .....	275
Dialogic .....	275
Natural Microsystems .....	275
NMS Conference Configuration.....	275
3 Best Practices for T1/E1 Cabling Management .....	277
<b>Index</b>	<b>278</b>



**TeleFlow**

**Part**



# 1 System Requirements

Generally speaking, any computer made in the last couple of years will be more than adequate for development. Below are some minimum guidelines (which are in some cases light enough that you can't even find them anymore) for both development and production servers, in addition to some suggestions for more productive systems. In each case below, keep in mind that anything "newer" or "more powerful" than what is suggested will be more than adequate (Exceptions: For TeleFlow Designer, be sure the operating system is listed as being supported. For TeleFlow Server, ensure the operating system is supported by the telephone hardware, and the software TeleFlow currently supports for it).

TeleFlow Designer, minimum system requirements

- Intel Pentium 4 PC (Basics: Monitor, sound card, speakers, a microphone for recording audio & testing)
- 80 GB Hard Drive
- Windows 2000 / XP Pro / Server 2003 operating systems.

TeleFlow Designer, effective full-time developer setup

- A good sound card (we've never gone wrong using Creative/Sound Blaster cards) is highly recommended if you will be doing your own recordings, and for testing using TeleFlow Simulator. Many computers come with integrated sound by default, and although the sound they produce is usually more than adequate, they often make lousy recordings.
- A higher quality microphone can make a difference if you are using your own recordings in production.
- 19"+ dual monitors, and a video card that supports them running side-by-side. (Flowcharting requires more space than other development...)
- 1 GB RAM

TeleFlow Standard Server (not including SR or TTS)

- 2 GHz CPU
- 1 GB RAM

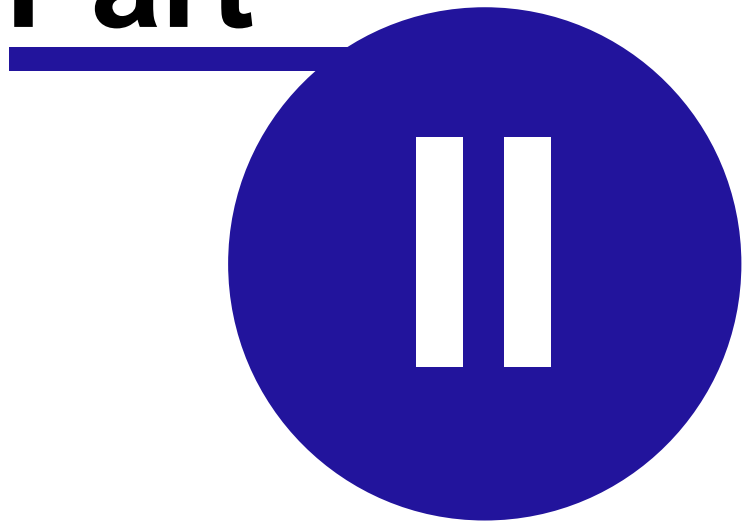
TeleFlow Media Server (with SR and/or TTS)

- 3 GHz CPU (or just about any dual-core or higher)
- 2 GB RAM
- 533 Front Side Bus or faster

Database applications will also require a database management system. MySQL for Windows is a good choice, although most any database accessible using ODBC will work. We recommend against using MS Access, as we have encountered numerous problems with Access and ODBC connections to it.

**TeleFlow**

**Part**



## 2 TeleFlow Installation

Depending on the type of telephony hardware being used, and whether the installation is being done with a TeleFlow CD or by downloading the required software, the installation process will vary. For instructions on how to install and configure TeleFlow with your hardware, read the section that corresponds to the hardware you are installing and the method of the install. First install the TeleFlow software.

### Installing The TeleFlow Software

The TeleFlow Server CD's installation process will install the necessary software for your telephony hardware.

Detailed download and installation information can be attained by clicking on the appropriate link for your hardware below:

#### **From a CD:**

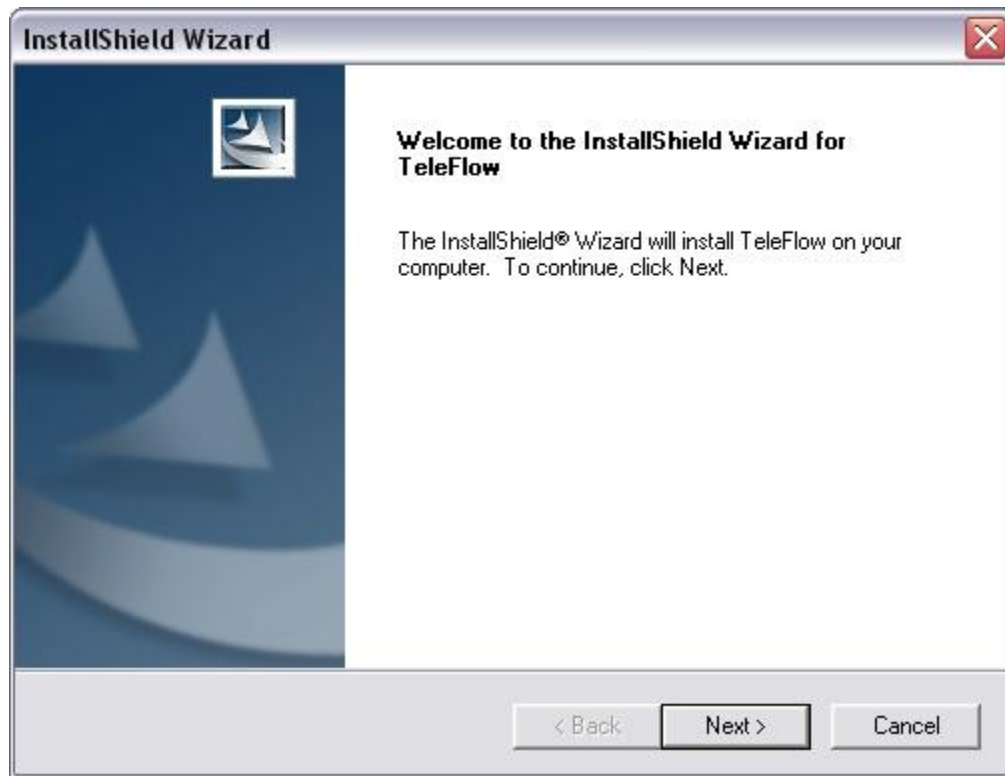
The TeleFlow Installer should start automatically when you place the TeleFlow Server CD in the drive.



Click "TeleFlow Setup" to start the TeleFlow Installation wizard. Follow the instructions of the wizard and answer its questions when prompted. Remember the location to which you install TeleFlow. It is recommended to use the default, which is C:\Program Files\TeleFlow.

**From a Download:**

Download the TeleFlow install package and run it.

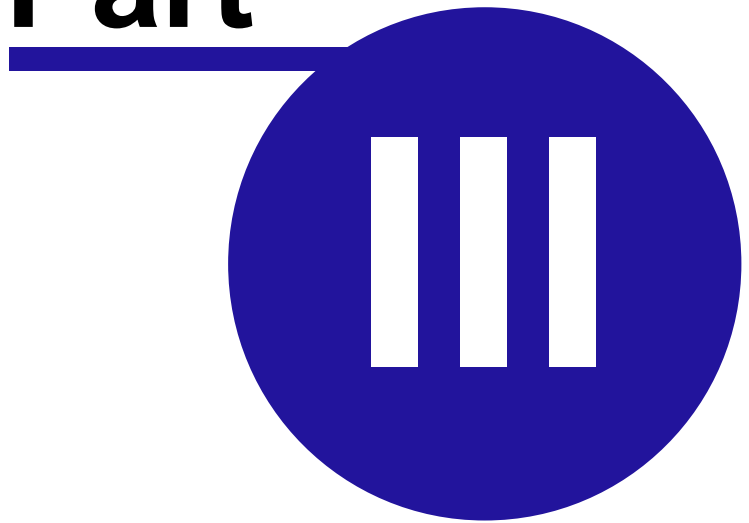


Follow the instructions of the wizard and answer its questions when prompted. Remember the location to which you install TeleFlow. It is recommended to use the default, which is C:\Program Files\TeleFlow.

The TeleFlow Installer should start automatically when you place the TeleFlow Server CD in the drive. Click 'TeleFlow Designer Setup' to get the Installation started. Follow the installation wizard to get the TeleFlow product suite installed. Remember the location to which you install it. (By default, this is C:\Program Files\TeleFlow)

**TeleFlow**

**Part**



## 3 TeleFlow Overview

TeleFlow is a full suite of software products that provide a user friendly interface providing everything necessary to design, test, implement, and maintain a voice application.

### Develop and Test an Application

#### TeleFlow Designer

The [TeleFlow Designer](#) is a development environment for building telephony applications using flowcharts. The Designer has flowchart steps encompassing all of the required logic for a telephony application. Just a few examples include: Answering a phone call, playing voice files, sending callers to information they want based on the input they provide through touch tones, and entering and retrieving information from databases. Applications built with the Designer can be tested using the [TeleFlow Simulator](#).

#### TeleFlow Simulator

The [TeleFlow Simulator](#) offers a simulation of how a TeleFlow application will run when placed in its production environment. This is how a TeleFlow application is tested by a TeleFlow application programmer. It does not actually answer or place calls, but it does simulate these, and most other activities a TeleFlow application has been built to perform. Once an application has been thoroughly tested with the Simulator, it is ready to be installed and tested using the [TeleFlow Server](#). It must first be set up and configured through [TeleFlow LineList](#) and [TeleFlow Config](#).

### Implement the Application

#### TeleFlow LineList

[TeleFlow LineList](#), or the application list builder, is required for building .TAL, or TeleFlow Application List files. The TAL file is what [TeleFlow Server](#) refers to in order to determine what application file(s) to run, how many (1 application is required for each IVR port), what telephony protocol(s) to use, what parameters to pass to the application, and several other options. The [TeleFlow Server](#) will use whatever TAL file is set by [TeleFlow Config](#).

#### TeleFlow Config

The [TeleFlow Config](#) utility sets the configuration values used and required by [TeleFlow Server](#), such as the TAL file to use, the directory to log to (what to log is determined in the TAL file), and the number of lines to log.

#### TeleFlow Register: Licensing

TeleFlow must be licensed from enGenic in order to be legally deployed. You are permitted, however, to test your application prior to registering TeleFlow. Consult [Licensing Requirements](#) for more information.

#### TeleFlow Server

The [TeleFlow Server](#) runs all of the TeleFlow applications (.TAP files) specified in the TAL file. The Server is responsible for starting the applications, communicating their output to the



---

necessary logs and to the [TeleFlow Monitor](#), re-starting applications that terminate incorrectly, and intercepting any communications sent to it from the TeleFlow Monitor.

## Monitor the Application

### TeleFlow Monitor

The [TeleFlow Monitor](#) is a utility for application monitoring . This status monitor displays the activity occurring on all TeleFlow applications currently running on the specified IVR computer. Activity is continuously monitored, then reflected onto the screen. The monitor indicates the status of each application through the use of status 'lights'.

**TeleFlow**

**Part**



## 4 Licensing Requirements

enGenic requires that you have a valid license to use TeleFlow, obtained from enGenic or one of its authorized distributors. Licenses are issued for a certain number of ports.

You may evaluate TeleFlow prior to purchasing licenses. In which case, TeleFlow will play advertisement messages, in addition to running your application. To disable these messages, follow the licensing procedure below.

### Licensing Procedure

The following procedures detail how to obtain and activate TeleFlow port licenses. These procedures assume that you have purchased or have an agreement to purchase) TeleFlow licenses. If you do not, contact [sales@engenic.com](mailto:sales@engenic.com) for assistance.

- Run the [TeleFlow Register](#) program.
- Start --> Programs -> TeleFlow -> Server Tools -> TeleFlow Register.
- Locate the "MAC Address" field.

Copy the contents of this field and paste it into an e-mail to [registration@engenic.com](mailto:registration@engenic.com)

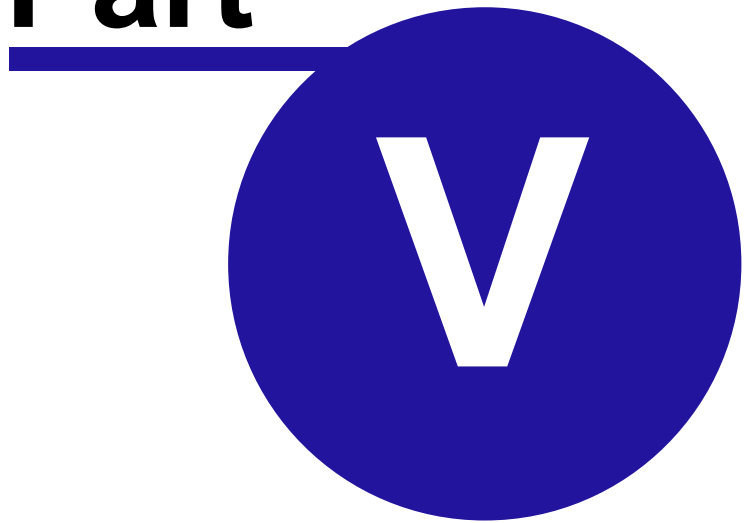
Our staff will respond with an email containing the values required for the remaining fields in the [TeleFlow Register](#) program.

### Third Party Licenses

Depending upon you application, certain [third party licenses](#) may also need to be acquired. For example, text-to-speech, speech recognition, and faxing require royalties to third party providers, and are therefore licensed separately. Note that the separate license requirements mean that you only have to purchase those third party components that your application needs.

**TeleFlow**

**Part**



## 5 1- Develop Applications with TeleFlow Designer

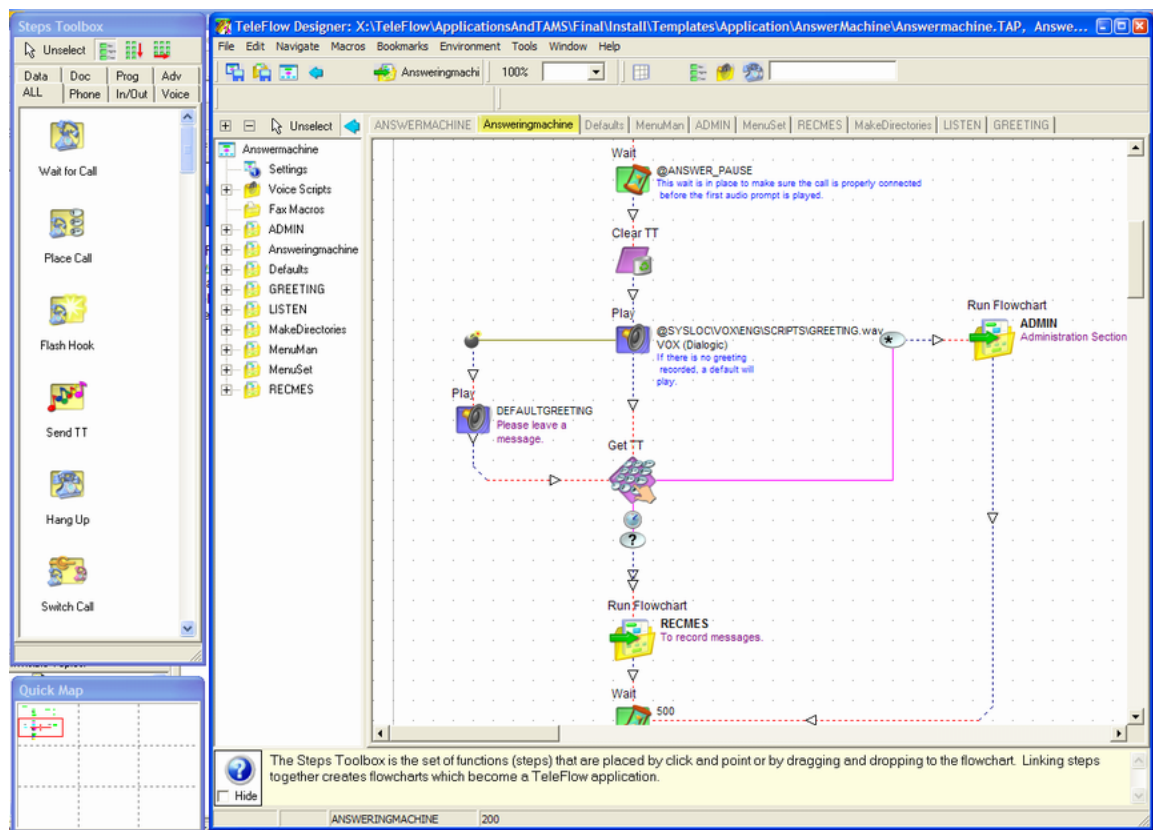
[What is the TeleFlow Designer?](#)  
[System Requirements](#)  
[Flowcharts](#)  
[Getting Started](#)  
[Getting Around in TeleFlow Designer](#)  
[Hints, Tips, and Tricks](#)  
[Step Definitions and Usage](#)  
[BasicScript](#)  
[Variables](#)  
[Voice Script Manager](#)  
[Voice Prompts](#)  
[FAQ's](#)  
[Stats and Reports](#)

### 5.1 What is the TeleFlow Designer?

TeleFlow Designer is a graphical telephony application development environment. A developer creates telephony applications by laying down steps to create a call flow in a Flowchart arrangement. Steps perform a variety of functions, from answering the phone, to gathering data from a database. Many flowcharts can be combined to create comprehensive telephony applications. The Designer manages all aspects of an application's development. This includes the application development environment, Voice Scripts and an on screen test engine, called [TeleFlow Simulator](#), that allows you to demonstrate how the application works.

TeleFlow Designer is used to develop the application. These can then be run in simulation on TeleFlow Simulator, or in production (with telephony hardware) on TeleFlow Server.

If you are new to TeleFlow Designer, or could just use a refresher on all the basics, the Getting Started chapter is a great place to start.



## 5.2 Flowcharts

The TAP Screen is the first window you will see after creating a new application or loading an existing one. This screen keeps track of and documents the various objects that comprise a telephony application. All Flowcharts and other components used by the system appear in this window. For information on adding or editing flowcharts, [click here](#). Two other windows that open automatically are:

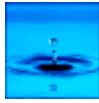
- [Quick Map](#)
- [Steps Toobox Window](#)

The TAP screen manages all of the flowcharts that you will create as you build the application. For an example of what a TAP screen may look like when you are finished, [click here](#).

The main TAP Screen has three icons already placed on it when you create a new application. The three icons each manage a distinct section of the telephony application. For a more detailed description of the icons, click on the corresponding link below:

- [Application Step](#)
- [Voice Script Step](#)
- [Starting Point Step](#)

## 5.3 Getting Started



This TeleFlow Icon is created during the installation process. It can be found in the TeleFlow folder under the Programs area of the Start menu. From the title page of TeleFlow, you can either create a new application, or open an existing one. If you have been previously working on an application, you will likely find it listed on the 'hot list' of applications listed under the File menu.

- [Creating a New Application](#)
- [Open an Existing Application](#)

The TAP Screen is the first window you will see after creating a new application or loading an existing one. This screen keeps track of and documents the various objects that comprise a telephony application. All Flowcharts and other components used by the system appear in this window. For information on adding or editing flowcharts, [click here](#).

Two other windows that open automatically are:

- [Quick Map](#)
- [Steps Toolbox Window](#)

The TAP screen manages all of the flowcharts that you will create as you build the application. For an example of what a TAP screen may look like when you are finished, [click here](#).

The main TAP Screen has three icons already placed on it when you create a new application. The three icons each manage a distinct section of the telephony application. For a more detailed description of the icons, click on the corresponding link below:

- [Settings Icon](#)
- [Voice Script Step](#)
- [Starting Point Step](#)

### 5.3.1 TeleFlow Basics

This section describes a number of important or key concepts that a TeleFlow application developer will need to understand in order to be successful with TeleFlow. It may not be important to absorb all of them right away (in fact, you might wish to skip this altogether, and walk through some of the demo applications first, before trying to understand this section), but ultimately, you will need to know most of the concepts presented here in order to develop all but the most basic of TeleFlow Applications.

**TeleFlow Variables:** TeleFlow variables are defined/denoted with the @ symbol. So, @VARIABLE could be a TeleFlow Variable name. Variables in Teleflow are NOT case sensitive, and are essentially all treated as UPPERCASE (in fact, they display that way in log output). At enGenic, we generally name our variables in UPPERCASE, because it makes the flowcharts easier to read. Since much of the information shown in flowcharts is in mixed case, this makes reading variables a little easier. You may find some of the samples will name variables with a leading lowercase "p". This is done when a variable is a parameter variable (passed into a flowchart), and is again to encourage easy readability. Obviously, this is all a matter of style. This has simply proven to work well for us.

**Variables and Data Types:** TeleFlow does not have strict data typing like you might be accustomed to in typical programming or scripting languages, or databases. Instead, TeleFlow treats all variables as strings, but will treat them as a different data type when the context of the operation being performed makes it sensible to do so. For example, if you use a TeleFlow Math step to add @VAR1 (contains "1") and @VAR2 (contains "1"), the result would be "2". If you attempt to simply append them together, they would make "11" instead.

**System Variables:** System Variables are read-only (generally) variables created and maintained by the system. They contain standard information that a TeleFlow Application developer might need available in an application, but cannot be readily accessed without TeleFlow's "help". (The date or time, for example) The chapter on system variables has help on the system variables available, and their names. Just like normal TeleFlow variables, system variable names start with the @ symbol. (Eg. @TIME)

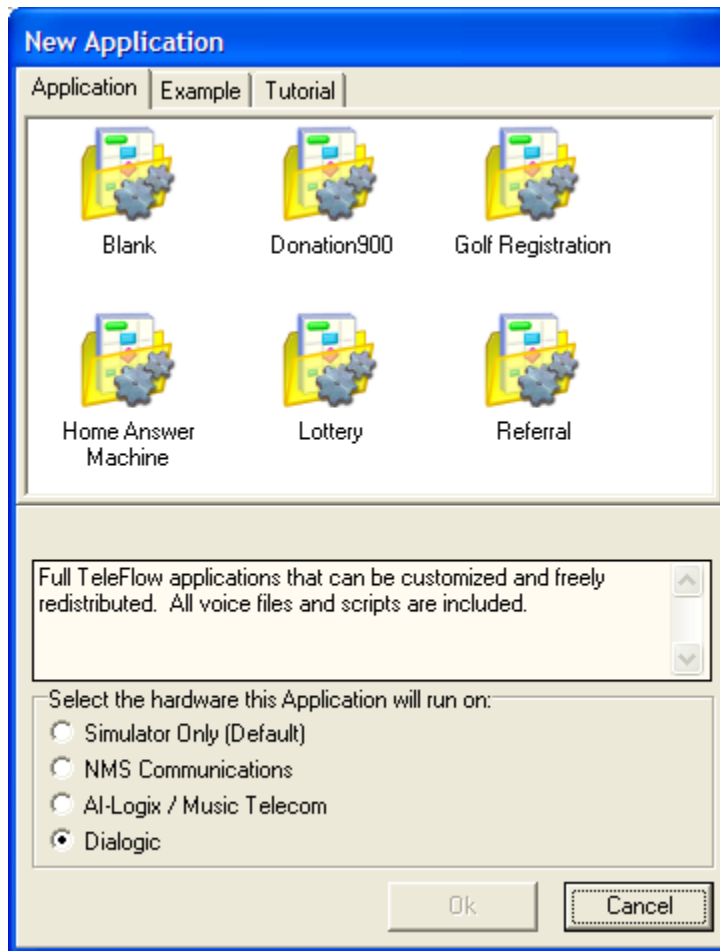
**Variable Scope:** Unless you specifically use the [Set Local Variable](#) step, or pass in a variable as a flowchart parameter on a [Run Flowchart](#) step, all variables created in TeleFlow are automatically global. Since TeleFlow is a flowchart environment, a variable (even a global one) does not exist until the application has "passed through" the step that creates it. For this reason, it is wise to have a single flowchart near the beginning of your application, that sets all the global variables you use throughout your application with a default value of some kind. Local variables are available only in the flowchart in which they are created. If you run another flowchart, or leave the current flowchart, the local variable falls out of scope.

### 5.3.2 Creating A New Application

#### Instructions:

- Select the **New Application** option from the **File** menu. The following dialog box will appear:





- Select either QuickStart start with an Application shell, or an application which closely resembles the application you want to create.
- A few seconds will pass as TeleFlow loads up the needed voice files and other associated files.
- You will be able to name the application and define the directory to save it to when you save the new application for the first time.

#### Notes:

A TeleFlow application, by default, includes a tree structure to save the associated files within. The TAM and Vox folders are created to store flowcharts and voice messages. This method is used because of the complexity of a TeleFlow application and the range of files that can be incorporated within it. For this reason, we recommend that all additional flowcharts created for the application be placed in the TAM directory.

For example, if the application VoiceMail was created in the directory IVR on C: drive, the following tree structure would result.

C:\VoiceMail\	VoiceMail.TAP	
	\tam	Location of the flowcharts (TAMs) associated with the
application (TAP).	\dbf	Database files
	\vox	All voice and script files

### 5.3.3 Opening an Existing Application

Select the **Open Application** option from the **File** menu.

The Open TeleFlow Application dialog box will appear (this box is similar to the standard Windows File Open dialog box).

Use the dialog box to find the application you wish to open. Note that only files with the .TAP extension will appear in the list. Once you have located the name of the application you want, double-click it, or select it and click OK.

A second way to open an application is to select the TAP file from the list of recently opened applications below the **Exit** option from the **File** menu. The application will be displayed in the application screen.

### 5.3.4 Adding Flowcharts to an Application


As you develop your application, you will need to add new or existing Flowcharts to it. Once you've placed and named the new flowchart, TeleFlow will automatically open that flowchart so that you may begin to place [Step icons](#) into it. The [Steps Toolbox Window](#) will now show an array of Step Icons that you may use. To edit flowcharts that are already placed on your TAP screen simply double-click on them with your left mouse button, and TeleFlow will open that flowchart for you.

- [Adding a New Flowchart](#)
- [Adding an Existing Flowchart](#)
- [Setting the First Flowchart](#)
- [Adding Step Icons to Flowcharts](#)

#### 5.3.4.1 Adding a New Flowchart

If you have just created a new application, this will be one of the first steps you will need to take during development. Every application needs to have at least one Flowchart as this is where the application flow, functions and logic is defined. The adding of Flowcharts must be done through the TAP Screen. If you are not already on the TAP Screen, click on either the first tab, or the home icon to select it. The TAP Screen is the screen that first appears when TeleFlow starts up. It is easily located in the tab list as it is the only flowchart tab that is in UPPERCASE letters.

#### Instructions:

- In the Steps Toolbox window, click once on the flowchart icon.  The mouse cursor will change to the Place Flowchart Icon.
- Click anywhere on the TAP Screen and the flowchart will be placed at that location.
- Give your new flowchart a name and define its location on your system.

#### Notes:

- Although it doesn't matter where you place the flowchart, you should attempt to keep some sort of system of placing 'similar' flowcharts with each other. By doing this, you may find it easier to locate specific flowchart if any future editing is necessary.

- Notes can be used to draw boxes around related flowcharts to further set them apart. The New TeleFlow Flowchart dialog box will appear (this is similar to the Window's File Open dialog box).
- It is best to leave the extension as .TAM. If you leave the extension off, TeleFlow will add the .TAM for you automatically. Click OK when you are done.

**See also:**

- [Adding an Existing Flowchart](#)
- [Setting the First Flowchart in an Application](#)
- [Adding Step Icons to Flowcharts](#)

### 5.3.4.2 Adding an Existing Flowchart

In order to add an existing flowchart to your application, you should ensure that the associated .TAM file has the same root directory that the .TAP file does. Then, in the [Step Toolbox Window](#) click once on the *Flowchart* icon. The mouse cursor will change to the *Place Flowchart Icon*. Click on the TAP Screen and the flowchart will be placed at that location.

**Instructions:**



- In the Steps Toolbox window, click once on the flowchart icon. The mouse cursor will change to the Place Flowchart Icon.
- Click anywhere on the TAP Screen and the flowchart will be placed at that location.
- When the New TeleFlow Flowchart dialog box appears, you should see the filename for the existing .TAM file.
- Click once on the filename in the dialog box, and then click on OK.
- The flowchart you just placed will be referenced to that .TAM file, and TeleFlow Designer will open the flowchart and allow you to begin editing it.

**Notes:**

- Although it doesn't matter where you place the flowchart, you should attempt to keep some sort of system of placing 'similar' flowcharts with each other. By doing this, you may find it easier to locate specific flowchart if any future editing is necessary.
- Notes can be used to draw boxes around related flowcharts to further set them apart. The New TeleFlow Flowchart dialog box will appear (this is similar to the Window's File Open dialog box).
- It is best to leave the extension as .TAM. If you leave the extension off, TeleFlow will add the .TAM for you automatically. Click OK when you are done.

**See also:**

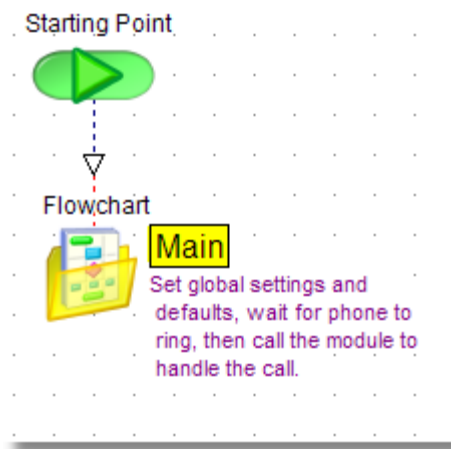
- [Setting the First Flowchart in an Application](#)
- [Adding Step Icons to Flowcharts](#)

### 5.3.4.3 Setting the First Flowchart in an Application

Every application requires a Starting Flowchart. This is the flowchart that will be initially run when the application is started using the TeleFlow Server or [Simulator](#). On the TAP Screen, find the



[Starting Point Step](#) and connect this icon to the flowchart that is to be the first flowchart in an application.



**The Starting Flowchart can easily be changed to connect to a different flowchart by breaking the link, and reconnecting to another flowchart.**

### 5.3.4.4 Adding Step Icons to Flowcharts

After you add a new flowchart to the TAP screen, the flowchart automatically opens to the 'Start' position for that flowchart. Placing step icons onto the flowchart is as easy as clicking once on the icon in the [Steps Toolbox Window](#), and then clicking on the spot in the flowchart where you want that icon to appear. Step Icons can also be dragged into position. Deciding which icon to place where is fully dependent on what the main function of that flowchart will be. Certain patterns can be followed in order to keep an IVR application easy to read and manage.

#### Instructions:

- In the [Steps Toolbox Window](#), select either the [Data](#), [Doc](#), [Prog](#), [Expert](#), [Phone](#), [In/Out](#), or [Voice](#) tab.
- Select the appropriate programming icon
- The mouse cursor will change to the Place Flowchart Icon.
- Click anywhere on the flowchart, and the icon will be placed there.

#### Notes:

Step Icons can be placed anywhere on a flowchart; however, if you follow no order as to where you place them, then the flowchart can quickly become unreadable. In order to avoid this, there are three things that may help. First, remember simply that you have nine full pages of blank space per flowchart. If you have a particularly large flowchart, spread out your step icons so that the flow from one icon to the next is easier to follow. Secondly, use the 'navigational' step icons to avoid jumbles of crossing flow paths. And lastly, when you find that you are using an identical series of step icons and scripts, consider making that series a

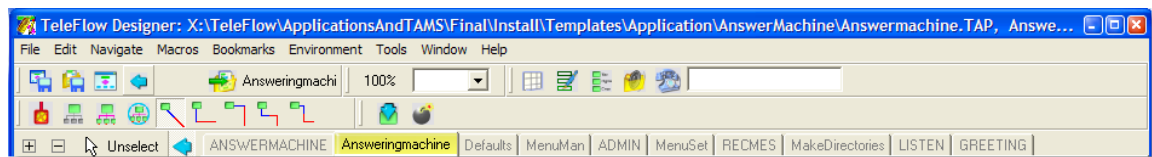
separate flowchart altogether, and then use the [Run Flowchart Step](#) when you need to run that routine. Several flowchart examples are listed below which show you some helpful methods of placing step icons in an 'easy to read' manner.

#### See Also:

- [Adding a new Flowchart](#)
- [Adding an existing Flowchart](#)
- [Setting the first Flowchart](#)

## 5.4 Getting Around in TeleFlow Designer

### TeleFlow Designer Toolbar



This button will 'Quick Save' the entire application.



This button will 'Quick Save' the currently open flowchart only.



This button will take you to the TAP Screen, which holds all the 'master' flowchart icons.



This button will take you to the previously opened flowchart. If you press it twice, it moves back two flowcharts, etc.



This button is only active if you have used the 'move backward' arrow button; whereas the previous arrow button will take you backward, this button will take you forward one flowchart. If you press it twice, it moves you forward two flowcharts, etc.

### Move To Flowchart Button

There is no Graphic for this button. When you click on a flowchart icon, the name of that flowchart will appear in this button. Clicking on this button will take you directly into that flowchart.

### View Control Icons

There are three view control icons that control the zoom level on a flowchart. The first zoom control button allows users to automatically re-size the flowchart to the 'default zoom percentage.' The second and third buttons allow developers to choose different zoom levels by using the pull down list, or the slide controls.



This button will bring up the [Quick Map](#) allowing you to navigate throughout the flowchart.



This button will bring up the [Step Properties Window](#) for the currently selected icon.



This button will bring up the [Steps Toolbox Window](#).



This button will bring up the [Voice Scripts Manager](#) window.



This button will automatically launch the [TeleFlow Simulator](#) so that you can test the developed application without any special hardware. Next to this button, there is an empty entry field. This field can be used to pass parameters to the



This button appears next to the 'Direct Link' button when you have an icon selected that is currently connected to another icon. This button will break that link between those two icons.

These buttons are the flow line control buttons:



Use them to bend the flows around corners to make your flowcharts easier to read. You can now press 1 through 5 on your keyboard to get the same flow.

These buttons will hide or unhide the action steps of the selected icon:



If the selected step icon has no 'hideable' action steps, these buttons will not be visually active.



This button will move to and select the next step connected to the currently selected step. If there are action steps available to move to (i.e. The [Menu](#) step has more than one possible path) the images of all applicable flow paths will also appear beside this button.

### Flowchart Tabs

As you create **flowcharts**, tabs are added to the Flowchart Tabs Line. By clicking on the tab that has the name of the flowchart, TeleFlow will take you directly to that flowchart. If you right-click the mouse button on the flowchart tab line, then a quick list of all of the flowcharts will appear in alphabetical order.

Note: When an application becomes large enough, this tab line will turn into two or more tab lines. Should you find that they are taking up too much space on the screen, then this line can be turned into a single slideable line by going into the settings option off of the [Applications Environment Settings](#) window.

#### 5.4.1 Flowchart Screen Menus

The menu options available from within an open application in TeleFlow Designer are detailed below. Each Menu is described by two columns of information, the left, or 'Menu Option' column is the label

that appears on the actual menu. The right, or 'Description' column provides the details about the menu option.

The menus are:

[File](#)  
[Edit](#)  
[Navigate](#)  
[Bookmarks](#)  
[Environment](#)  
[Tools](#)  
[Window](#)  
[Help](#)

## File Menu

Menu Option	Description
Open Application	Allows you to open a previously created TAP application. If another application is currently open, TeleFlow will prompt you to save any unsaved changes before opening another application.
New Application	Allows you to create a new TeleFlow (.TAP) application. This menu option will allow you to choose the type of application you want to create. If an application is currently open, TeleFlow will prompt you to save any unsaved changes before creating a new application.
Save Application	Saves changes made to the current application, including all flowcharts.
Save Application As	Allows you to save the application under a different name and/or in a different location.
Close Application	Closes the application, and prompts you to save it only if unsaved changes have been made. As opposed to the exit command, this will keep TeleFlow running.
Print Application	Prints the entire IVR application. This includes the TAP screen and all .TAM flowcharts. The flowchart grids are printed at a scale of one page per grid. This print option is extremely useful for testing a complete IVR application as you can print and highlight flows as you test them. Highlighting the printed flows as you test them ensures that no flows are inadvertently missed.
Print Flowchart	Prints the current flowchart. A full flowchart contains nine pages, or grids. When you select this print option, it will ask you whether you want to print all nine grids on one page (full view), or print each grid on a separate page (detailed view).
Print Screen	Prints the viewable area of the current flowchart, including the flowchart 'tabs' at the top. This is similar to printing a screen capture except that the pull-down menus and title bar are not included.

Exit	This menu option will close TeleFlow. If unsaved changes have been made to the open application, TeleFlow will prompt you to save the file before closing.
1-20	At the bottom of the File menu, a list of up to nineteen recently used applications is kept, most recently opened first.

## Edit Menu

Menu Option	Description
Undo Ctrl+Z	Allows you to undo the last action taken.
Cut Ctrl+X	Copies the selected icons from the screen to a temporary buffer and removes them from their current position on the flowchart, allowing you to paste them elsewhere.
Copy Ctrl+C	Copies the selected icons into a temporary buffer without removing them from their current position, allowing you to duplicate the series of icons elsewhere.
Paste Ctrl+V	Pastes the information currently in the buffer to the location on the screen that you have specified.
Locate Same Steps Ctrl+L	Locate all steps in the application that are the same as the currently selected step. Populates results into the List All Steps window.
Search Steps for Text Ctrl+F	Opens the Search Text window, which allows you to search the entire application for fields or comments containing text you indicate.
Delete Delete key	Deletes the currently selected step.
Step Properties Ctrl+E	This will open the step properties window for the currently selected icon.
Hide Action Steps Ctrl+H	Hides a currently selected action step, or all action steps for a selected step.
Unhide Action Steps Ctrl+U	Unhides actions step for a selected step.
Break Link Ctrl+B	When a single icon is selected on the flowchart, this option appears under the edit menu. It will only appear if the selected icon is currently connected to another step icon. When used, it will break the link between the selected icon and the other step icon that it is connected to.
Direct Link	This option creates a straight line between two icons. If there is no currently selected icon, this menu option will not be active.



Vertical 'L' Link	This option creates an 'L' shape in the line connecting two step icons. The flow line will travel up/down from the first icon, and then left/right to the next icon. In some complicated flowcharts, changing the direction of a flow line will make flow lines easier to follow and read.
Horizontal 'L' Link	This option also creates an 'L' shape in the line connecting two step icons; however, the flow line travels horizontally first, and then vertically.
Vertical 'S' Link	This option creates two bends in a flow line connecting two icons, traveling vertically first.
Horizontal 'S' Link	This option creates two bends in a flow line connecting two icons, traveling horizontally first.
Auto Connect Ctrl+A	This option will automatically connect the next icon placed on the flowchart to the current step. A yellow box appears around the connect icon.

## Navigate Menu

Menu Option	Description
to Top Left Home	Each flowchart is made up of nine 'pages'. Using this command will take your view to the top left area of the flowchart.
to Step Number	Each step icon is given a step number when it is placed upon a flowchart. Using this command allows you to go directly to any step number. This command is most used for debugging purposes when used in conjunction with the <a href="#">TeleFlow Simulator</a> . As the Simulator runs the current application, the flowchart and step number is displayed for each function. If the Simulator should fail as it runs the application, the step information appearing in the messages box can be used to move to the step.
to Selected Step	This command is used in an instance where you are navigating through a flowchart, and you want the view to 'snap back' the currently selected step.
to Next Step Ctrl+M	This command will select the next step connected by the current step in a flowchart.

## Bookmarks Menu

Menu Option	Description
Set Bookmarks (1-10) Ctrl+(F1-F10)	Assigns a bookmark to the application bookmark file. A bookmark 'remembers' a flowchart and screen position within the flowchart, and allows you to jump to a bookmarked location at any time, by either selecting a Bookmark from the Bookmarks menu, or using the quick keys assigned to them.

Bookmark (1-10)  
Shift+(F1-F10)

List of bookmarks, with the flowchart name the bookmark is for in parentheses. Selecting one of these takes you to the bookmarked location in the flowchart.

## Environment Menu

Menu Option	Description
Quick Map always on Top	Putting a check here will ensure that the <a href="#">Quick Map</a> does not get hidden behind other windows.
Step Properties always on Top	Putting a check here will ensure that the <a href="#">Step properties</a> window does not get hidden behind other windows.
Steps Toolbox always on Top	Putting a check here will ensure that the <a href="#">Steps Toolbox</a> does not get hidden behind other windows.
Set current as default Percent	When you first open an application, by default they are viewed at the 100% zoom level. You can change this default by setting the zoom level that you wish to use, and then executing this option.
Options...	Brings up and allows you to change the <a href="#">Application Environment Settings</a> .

## Tools Menu

Menu Option	Description
Test with Simulator	Tests the currently open application using the <a href="#">TeleFlow Simulator</a> .

## Window Menu

Menu Option	Description
Quick Map	Snaps the <a href="#">Quick Map</a> to the front of all other windows.
Step Properties	Brings up the <a href="#">Step Properties</a> window for the currently selected icon. If the window is already open, but hidden behind another window, then this menu option will snap the Step Properties window to the front.
Steps Toolbox	Snaps the <a href="#">Step Toolbox</a> window to the front.
Voice Script Manager	Brings up the <a href="#">Voice Script Manager</a> .
Fax Macro Manager	Brings up the Fax Macro Manager.

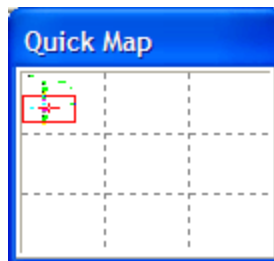
## Help Menu

Menu Option	Description
-------------	-------------

Contents		Opens help window to the main content page.
Step Help	F1	Opens help window on the topic page for the currently selected step.
Welcome Screen		Brings up the 'Welcoming to TeleFlow Designer' Screen.
Overview Screen		Brings up the TeleFlow Overview, which provides some general information about the different windows in TeleFlow Designer.
Activate Assistant		Activates the TeleFlow Assistant, which provides some basic help while you are using TeleFlow Designer, describing different windows, steps, and features as you use them.
Contact Support & Sales		Link to enGenic web site contact page.
Review System Log		Opens the system log, which provides version info, in addition to an application activity/progress log.
About		Copyright and Version information about TeleFlow.

### 5.4.2 Quick Map

A Flowchart comprises a very large workspace (nine printed 8.5" x 11.0" pages to be exact). It is impossible to display the entire flowchart on the screen at one time and still be able to read it. The Quick Map shows you the entire nine pages at a glance. The red box represents the amount of the entire flowchart that is visible on the Flowchart Screen at any one time. By pressing the mouse button on the Quick Map and moving, you can move the red box to change the view location of the Flowchart Screen.



### 5.4.3 Steps Toolbox Window

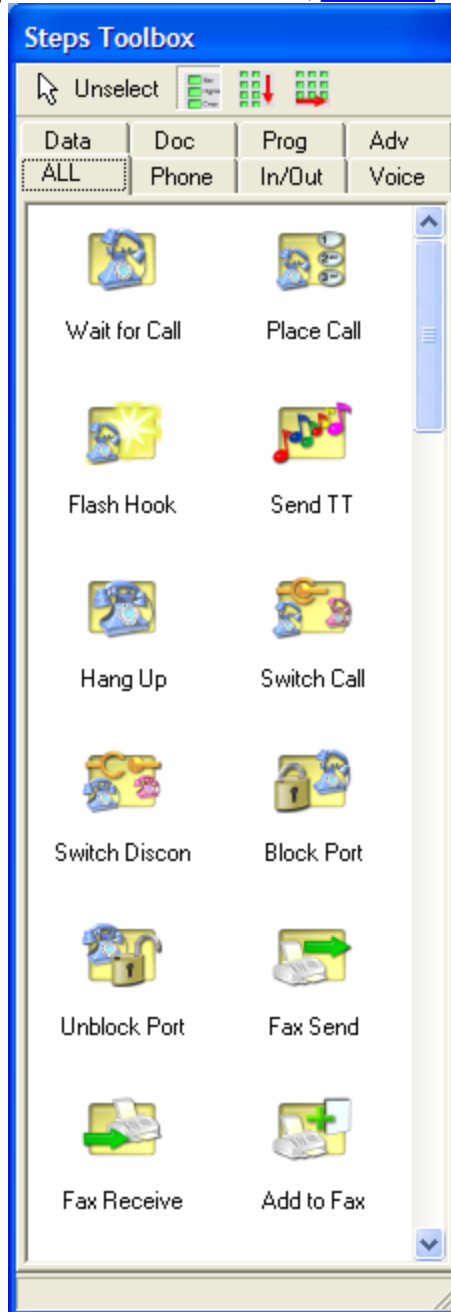
The Step Toolbox Window contains all of the Step Icons (a step can be thought of as one command that the TeleFlow Server or [Simulator](#) is capable of performing) that can be used within a [flowchart](#). When TeleFlow displays the main TAP screen, the Step Toolbox Window displays only two steps: a Flowchart step, and a Notes step. Those two steps are the only two steps that may be placed onto the TAP screen.

To add a step to a flowchart, click on the Flowchart step in the Steps Toolbox Window, and then click on the flowchart area at the location where you want to place the step. While you have a step selected, the mouse icon will change to the Step Placement Cursor. To place more copies of the same icon, simply continue clicking on the locations where you want the icons to be placed.

Once placed, a step can be clicked on to bring up its associated [Step Properties Window](#). Each type

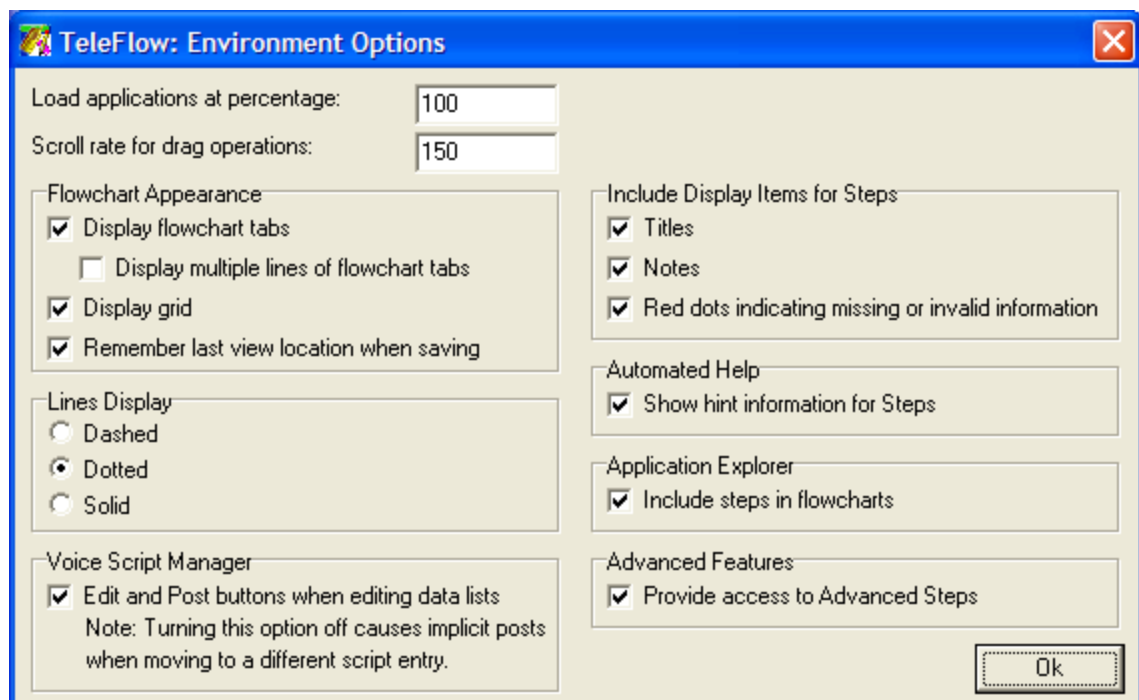
of step has its own distinct [Step Properties Window](#), which is a dialog box that allows you to configure the step to do exactly what you want. For example, an [Wait for Call](#) step can be configured to wait for a certain number of rings before answering an incoming call.

For a complete listing of all Step Icons and their functions, [click here](#)



#### 5.4.4 Application Environment Settings

The application environment settings window can be accessed from the Environment Menu in TeleFlow Designer.



### Load applications at percentage

Controls what zoom factor is used for opening future applications.

### Scroll rate for drag operations

Controls how quickly the screen moves when you drag objects to an 'off-screen' location.

### Flowchart Appearance

#### Display multiple lines of flowchart tabs

In large applications the sheer number of flowcharts can add so many tabs to the toolbar that you lose half of your application space to them. If this option is unchecked, it ensures that only one line of tabs is visible, and scroll arrows are used to access the ones that are off-screen.

#### Titles

Adds or removes the step icon titles. When flowcharts contain large amounts of icons, removing the icon titles can increase the readability of a flowchart.

#### Notes

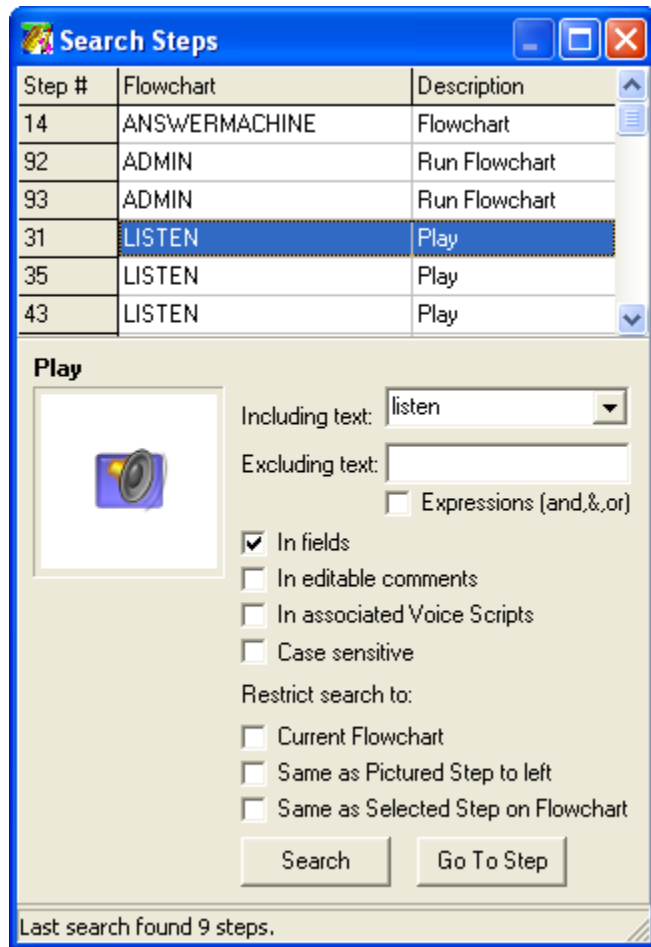
- This command acts as the *titles* command does, only it adds/removes the notes from the step icons. The notes are not deleted from the icon, they are just not displayed on the flowchart.

#### Red dots indicating missing or invalid information

- Controls the debugging mode where icons missing information are automatically shaded red.

### 5.4.5 Search Steps for Text

The Search Steps for Text window, available from the TeleFlow Designer menu option - "Edit >> Search Steps for Text", or by pressing Ctrl-F while in the Designer - allows you to search an entire TeleFlow Application (.TAP) for text that appears in step fields or comments.



#### Step #

The step number - unique to the flowchart - in which the step was located.

#### Flowchart

The name of the application Flowchart that the step was located in.

#### Description

The name of the step that the search located.

#### Step Picture and Name

In the left hand of the search window, beneath the list of search results - if any - is a picture box. This picture and label indicate the currently selected step in the search results window.

**Including text**

The text to search steps in the application for. NOTE: The drop down list stores 9 most recent searches, and all of the settings the search used.

**Excluding text**

Text to ensure DOES NOT come up in the results list. (If the 'including text' is found in a step that also contains the 'excluding text', the step will not show up in the search results)

**Expressions (and,&,or)**

Indicates whether or not (checked or unchecked, respectively) to use expressions in the Including and Excluding text entries. If this setting is on, you can search for step that include several different text values, or one of a group of values, by using the logical operators: And and Or. (ie. Searching for 'listen OR delete' while this checkbox is checked will search for steps with either of these values, while a search for 'listen and delete' while it is checked will search for steps containing BOTH of these values)

**In fields**

Indicates whether or not the search should check for the indicated text in step property fields.

**In editable comments**

Indicates whether or not the search should check for the indicated text in editable comments in steps. (The Notes step - among others - only has a comment box)

**In associated Voice Scripts**

Indicates whether or not the search should check for the indicated text in associated voice scripts. (A play step, for example, contains a script code, but not the actual associated script text. By checking this box, you are indicating you want to locate steps that have codes associated with script text matching your 'Including Text' entry)

**Case sensitive**

Indicates whether or not the search should check for the indicated text with matching case ONLY. (ie. A search for 'DOG' wouldn't find a step containing 'dog')

**Restrict search to**

Checkboxes indicating whether or not to restrict the search for the indicated text to one or more of the following criteria:

- **Current Flowchart:** Search in the current flowchart only
- **Same as Pictured Step to left:** Search for the type of step (ie. a Play step) in the picture box only
- **Same as Selected Step on Flowchart:** Search for the type of step (ie. a Play step) that is currently selected (Selected: outlined in red) on the flowchart

**Status Bar/Search Count**

The status bar in this window shows the count of the most recent search.

**Search Button**

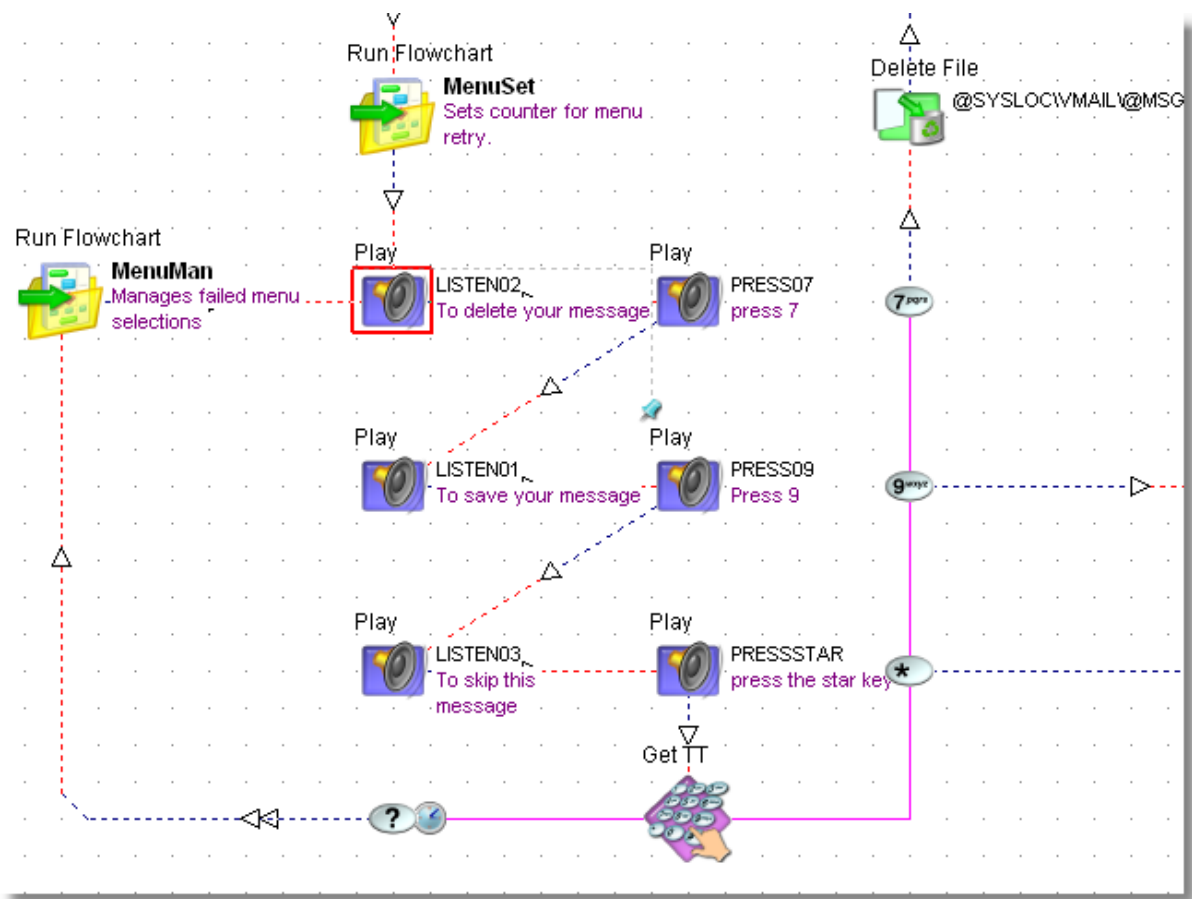
Runs the search.

### Go To Step Button

Goes to the step currently highlighted in the search results window. Double-clicking the step/row in the results window will also go to the step.

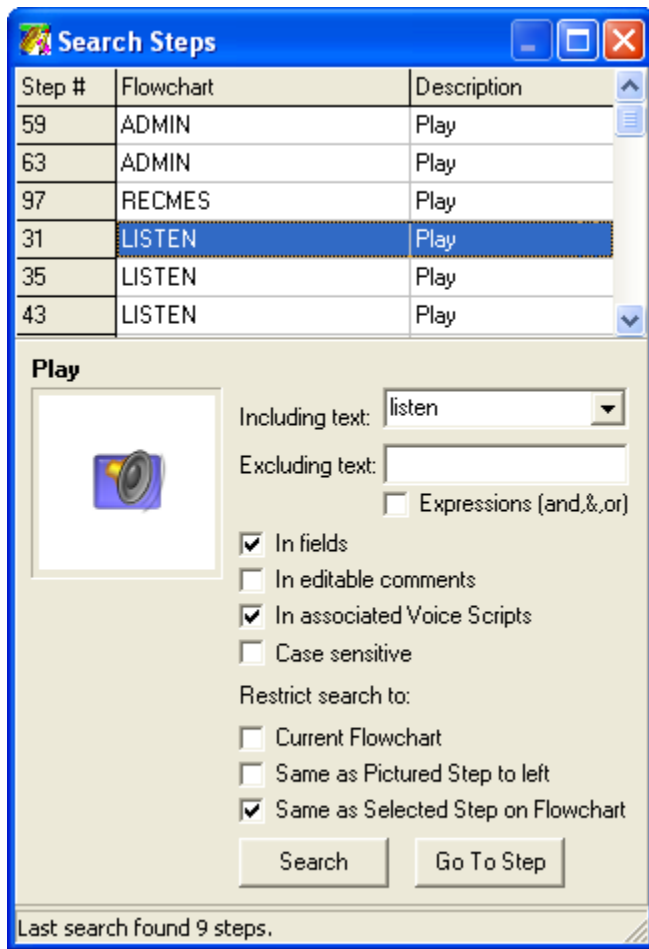
The following example figures use the AnswerMachine demonstration application to show some of the ways in which Search Steps for Text can be used.

Click on the play step and then use CTRL + F to bring up the search.



This window (Search Steps) will appear. Click the "unassociated Voice Scripts" to check the voice script text as well. Type in the text of the play step you wish to find. For example, if you had multiple Good-bye recordings, type in good-bye and they will all come up. You could use the Excluding text parameter to narrow the search down.





## 5.4.6 Hot Menus

There are three different 'hot-menus' available on the flowchart screen. Two of them are available by clicking the right mouse button on a step or on a blank flowchart space. The third menu is available when you select an area or a group of icons.

- [Step Options Hot Menu](#)
- [Selected Area Hot Menu](#)
- [Tablet Options Hot Menu](#)

### 5.4.6.1 Step Options Hot Menu

If you right-click on a selected step icon, then a menu pops up with a number of commands that you may perform on that step-icon. Depending on which step icon you click on, any of the following commands will be included in the menu that is presented to you:

Menu Option	Description
-------------	-------------

Edit Properties	Open the step properties window for the selected icon.
-----------------	--

Unhide Action Steps	If there are hidden action steps for the selected icon, this command will appear on the menu. Clicking on this command will simply make the action steps visible again.
Gather Action Steps	Pulls unlinked action steps back into the position that they were in when the step was originally placed.
Hide	Hides in-connected or unused action steps to reduce flowchart clutter and force flow to other action steps.
Quick Hide	Used where a step icon has more than one action step that is not connected to another step. Quick hide will automatically hide all unconnected action steps.
Copy	Copies the selected icon to the clipboard.
Delete/Confirm	Deletes the selected icon. Must be confirmed.
Break Link	Breaks the connection between the selected icon and the next step.
Go to Next Step	Follows the flowchart line and selects the next connected step.
Flow Line Controls	Changes the style of line connecting the next step icon.
Help	Begins context sensitive help.

#### 5.4.6.2 Selected Area Hot Menu

After left-clicking and dragging a selection box around a group of icons, you are presented with a hot menu that allows you to perform a number of actions on the selected area. The commands and their functions are as follows:

Menu Option	Description
Relocate	Allows you to shift the location of the selected icons while keeping all connections intact.
Cut	Copies the selected area into the buffer and then clears the selected area from the flowchart.
Copy	Copies the selected area into the buffer but does not clear the selected area from the flowchart.
Delete/Confirm	Deletes the selected area. Must be confirmed.
Isolate	Any flow lines coming into or going out of the selected area are severed, while any flow lines between two icons within the selected area are kept intact.
Shatter	Any flow lines within the selected area are severed, while any flow lines coming into or out of the selected area are left intact.
Isolate/Shatter	Severs all flow lines associated with the selected area.

#### 5.4.6.3 Tablet Options Hot Menu

If you right-click on an area of the flowchart that is not occupied by an icon, then a hot menu will open with the following commands:

Menu Option	Description
Insert Space	In instances where you need to add additional steps in a flowchart, but there is now room, the 'Insert Space' command will allow you to shift the existing steps in order to create space for the new ones. The cross hair that appears will allow you to shift any steps below and/or right of your current cursor position.
Paste	Pastes the contents of the cut/copy buffer onto the flowchart at the current cursor position.
Go to Top Left	Shifts your view to the top left corner of the current flowchart.
Go to Selected Step	Centers your view on the currently selected step.
Go to Step #	Selects and centers your view around the step number that you input. The step numbers are displayed on output from the TeleFlow Server and Simulator.

## 5.5 Hints, Tips, and Tricks

[Setting up Menus](#)  
[Managing Repetitive Voice Scripts](#)

Press 1 for option X; press 2 for option Y'

Shows an example of how to best re-use repetitive pieces of scripts.

[User Defined Variable](#)

Shows an easy way to set all of your variable defaults

[Separating and Calling a Repeating Series of Icons](#)  
[Navigational Labels](#)  
[Connect Point](#)

Separating and 'calling' a repeating series of icons

How to keep flowcharts legible using labels

Allows several paths to become one common path

### 5.5.1 Isolating a Repetitive Series of Icons into a Separate Flowchart

Throughout the development of IVR applications, you may find that you use a specific pattern of icons and scripts at several different locations. Instead of duplicating the pattern every time you need it, is easier to move the pattern into a separate [flowchart](#). Then, whenever you require those icons and scripts, simply use the [Run Flowchart](#) step to execute that series of icons. When the TeleFlow Server or [Simulator](#) reaches the 'Finish' icon in the flowchart, it will automatically return to the last position in the previous flowchart.

As an example, we will show you how to isolate a menu management routine that repeats menus when the caller inputs an invalid key, or does not input anything at all.

Most IVR systems offer at least one menu to the caller (press 1 for option A, etc). If the caller enters an invalid option (they press 9, but it wasn't offered), or if they simply don't press anything (the time-out is reached), then the IVR needs to offer the menu again. Most often, an IVR will give the following instruction:

*"We're sorry, we did not receive a valid input, please listen to the options again, and enter your selection at any time."*

Along with the voice script, the IVR should be developed to use an arbitrary variable to count how many times the menu has been repeated. On average, most menus should repeat at least once, and as many as 3 times before playing a script that says something to the effect of:

*"We're sorry, we still have not received a valid input, please try your call again later, or call (555) 555-5555 for technical support, thank you."*

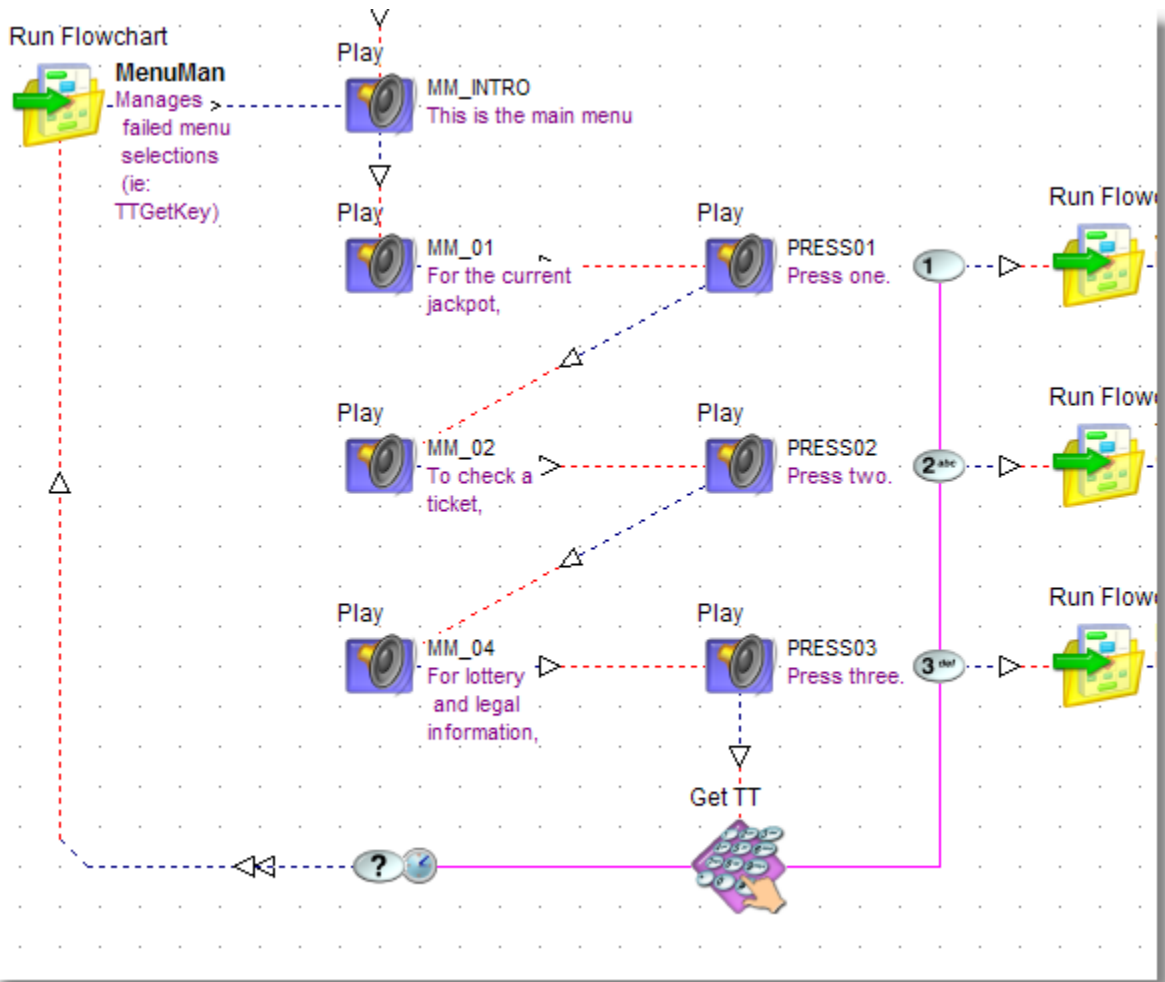
By keeping track of the number of repeats, the IVR doesn't get stuck into an endless repetition of the same menu. After a specified number of attempts, the IVR will hang up and reset that line back to 'wait for call.'

Because input is normally required in several spots throughout an IVR application, the process of repeating the menus should be made into a separate flowchart. This negates having to re-map the series every time it is required. The following three links will show you how such a process would be incorporated into a menu. Arbitrarily, we use the variable @COUNTER.

### 5.5.2 Managing Repetitive Voice Scripts

As you create an [IVR](#) application, you will be placing many steps that will speak information to the caller. As you get used to creating applications, you will find it necessary to come up with voice file naming standards that you can follow. For example, because the phrases 'Press 1 now' or 'Press 2 now' are normally used repeatedly in an application, you may find it useful to separate these script files from the 'reasons' why a caller should press one or two. That way, whenever you want to give a menu for the caller to choose from, you can re-use the 'press this digit now' scripts. Other advantages become evident quickly when using this approach. For example, if you need to add a menu choice at the beginning of a list of menu options, the original menu options can easily be bumped up. An example follows where the 'press this digit now' voice files are separated:

Example:



### 5.5.3 Setting Icons Up So The Flow is Easy to Follow

Step Icons can be placed anywhere on a flowchart; however, if you follow no order as to where you place them, then the flowchart can quickly become unreadable. In order to avoid this, there are three things that may help. First, remember simply that you have nine full pages of blank space per flowchart. If you have a particularly large flowchart, spread out your step icons so that the flow from one icon to the next is easier to follow. Secondly, use the 'navigational' step icons to avoid jumbles of crossing flow paths. And lastly, when you find that you are using an identical series of step icons and scripts, consider making that series a separate flowchart altogether, and then use the [Run Flowchart Step](#) when you need to run that routine. Several flowchart examples are listed below which show you some helpful methods of placing step icons in an 'easy to read' manner.

The following example show how a series of [Play](#) steps are set up so that the options are easy to read and the flow from one option is more apparent. Also, the [Menu](#) action steps are spread out so that they appear beside the menu option that they represent.

### 5.5.4 Using Labels

When you have two areas of a [flowchart](#) that need to be connected together, but doing so will cause flow lines to cross and possibly become confusing to follow, [Label](#) steps can be used to negate the crossing of lines. Using labels is a two-step process. The first step is to place a [Label](#) where you want to end up. The second step is to place a [Go To Label](#) step where you are jumping from.

Note that the label that you are jumping to must be either in the flowchart currently running, or in a flowchart that has been previously opened during the same call.

### 5.5.5 Using the Connect Point Step

The Connect Point icon is used to redirect a flow or merge two or more flows that are going to the same next step.

[Using the Connect Point to Re-Direct a Flow](#)

## 5.6 Steps Toolbox

The Steps Toolbox window contains all of the TeleFlow steps, organized into convenient tabs. A TeleFlow step performs a particular task (or tasks), such as playing an audio file to a caller. TeleFlow flowcharts are created by linking steps together to form an application flow, or call flow.

This chapter topic provides some help on the Steps Toolbox itself, action steps and step properties. The chapter provides individual step help on all the steps available for use in a TeleFlow Application.

[The Steps Toolbox Window](#)  
[Action Steps and Step Connections](#)  
[Step Properties](#)  
[TeleFlow Steps Table of Contents](#)

### The Steps Toolbox Window

Adding steps to your application/flowchart simply requires locating the correct step in the Steps Toolbox window, clicking it, and dropping it on the flowchart space where you want it. (Drag-and-drop is also supported)

The Steps Toolbox itself has some options for how the steps are organized and displayed. These control whether to view large or small (i.e. compact images) step pictures, how to order them, and whether to sort by step type or alphabetically. The best way to find out which works for you is to simply experiment with them until you find the settings you like.

The steps in the Steps Toolbox are organized into category tabs to make them easier to locate. If you aren't sure what step to use to do something, but know you want to answer the phone, you could check the "Phone" tab to see if the task you are trying to perform is covered by one of the steps there.

The tabs available in the Toolbox are different in the TAP Screen than they are when editing a flowchart. The tabs and what you can generally expect to find in them follow.

#### TAP Screen

Standard: The steps here allow new flowcharts and comments to be added to the application.

Templates: The Templates tab has template flowcharts that can be added to an application. Each of these performs tasks not handled by TeleFlow steps that are often required in TeleFlow applications, such as creating a unique filename.

#### Flowcharts

ALL: Every step available to use in TeleFlow Designer is available from the ALL tab. Recommended for advanced users who know exactly what they're looking for.

Adv.: The Adv. tab is for advanced steps.

Audio: Contains steps related to audio files or sound, such as playing an audio file to a caller, or recording one.

Data/XML: Contains steps for working with databases and xml documents.

Doc.: Contains documentation steps. Documentation steps are those that help with application documentation in TeleFlow Designer, and those that can help with custom reports and logging by writing to external files.

Input: The Input tab steps are all for gathering and reacting to caller input by touch tones. The SR tab has similar steps that work for Speech Recognition.


Phone: Contains steps relating to using a phone, such as dialing digits or placing an outbound call.

Prog.: The Prog. tab contains TeleFlow Programming steps. Programming steps in TeleFlow cover logical operations, file manipulation, variable handling, global events, labels, scripting, logging, running flowcharts/programs/apps, and calculations.

SR: Contains steps for using Speech Recognition in applications.

Web: The Web tab steps are steps relating to web/network/internet related tasks, such as sending email or downloading a web page.

## Action Steps and Step Connections

Almost every step has at least one connector , or dangler, attached to it. The connector, when dragged and dropped onto another step, connects the step it is attached to to the step it is dropped on. This creates a flow line from the connector's step to the step it is dropped on. When a step has only one connector, this is its SUCCESS connector, and the flow it creates between two steps is the SUCCESS path (i.e. When the step does what it was supposed to do without error, it follows this path to the next step).

In addition to the success connector, steps may have "Action Steps" hanging off of them, and/or available to be used. Action Steps provide a way to create optional call flow when a step leaves for a reason other than successfully completing to task for which it exists (i.e. They allow you to create an alternative action to take from that of the success path). Some steps don't have a success path at all, instead having action steps related to actions taken by the caller, or based on logical decisions within the application.

Some steps don't show (all) their action steps when added to an application, because they aren't always used. The Play step, for example, has a Fail action step that can be used to set a Fail path so that an application can do something else if the file to play doesn't exist. This is optional because a missing file is normally an error the application can't do anything about.

To show action steps for a step that doesn't show them by default, right-click the step in the flowchart space, and select the "Unhide Action Steps" option.

## Step Properties

Many steps have properties that must be filled in before they can perform the task they are intended for (Eg. A Play step requires script code or audio file references before it can play

anything). Others have optional properties that can be filled in to take advantage of certain features, or to change how the step behaves. Some few steps have no properties that affect how the step behaves, because the task it performs doesn't have any options (Eg. Beep plays a beep tone).

All steps have comments sections in the bottom-right hand of the Step Properties window. This section is for user-defined/application developer comments. The comments are visible in the flowchart space, in blue, to the right of the step, beneath any automatic documentation the TeleFlow Designer provides for the step.

### 5.6.1 TeleFlow Steps

This chapter describes each Step that is available for placement in a TeleFlow flowchart. Many of the examples that are included for each Step refer to [variables](#). Variables are denoted by an @ symbol.

The help for each step uses the following basic layout. Some of the categories described below won't be shown if they are unnecessary to completely describe the step:

Description of what the step does.

PROPERTIES: Detailed help on every step property.

ACTION STEPS: The action steps that are available for the step, and what they mean for it.

RELATED STEPS: Other TeleFlow steps that are similar in nature, or work well conjunction with the step.

SUPPORTED TELEPHONY HARDWARE PLATFORMS: For steps requiring telephony hardware, this section lists which hardware is supported.

SPECIAL HARDWARE CONSIDERATIONS: This section explains limitations or special issues of note when using the step with particular hardware.

#### TABLE OF CONTENTS

The following chapters and topics are organized just like the Steps Toolbox: Each chapter represents one of the tabs in the Steps Toolbox, with the following exceptions:

- The ALL tab isn't recreated here.
- The START and FINISH steps chapter is a special section on the "automatic" steps that every flowchart has.
- The Action Steps chapter is a special chapter describing what action steps a step might have, and what each is generally for.

[START and FINISH steps](#)

[Starting Point](#)

[START step](#)

[FINISH step](#)

[Action Steps](#)

[Busy](#)

[Connector Icon](#)

[Failure Action](#)

[Invalid Action](#)

[Telephone Buttons](#)

[Time-Out Action](#)

[Unrecognized Voice-Action](#)



[Yes / No Action Steps](#)  
[Phone Steps](#)  
[Block Port](#)  
[Open Conference](#)  
[Join Conference](#)  
[Leave Conference](#)  
[Close Conference](#)  
[Disable Reorder](#)  
[Enable Reorder](#)  
[Add to Fax](#)  
[Fax Clear](#)  
[Fax Receive](#)  
[Fax Send](#)  
[Flash Hook](#)  
[Global Event](#)  
[Hang Up](#)  
[Pick Up](#)  
[Place Call](#)  
[Register SIP URI](#)  
[Send TT](#)  
[Set Media for VoIP](#)  
[Set Reorder Tone](#)  
[Switch Call](#)  
[Switch Disconnect](#)  
[Transfer Call](#)  
[Unblock Port](#)  
[Wait for Call](#)  
[Input Steps](#)  
[Clear TT](#)  
[Get Letter](#)  
[Get TT String](#)  
[Menu](#)  
[Pending TT](#)  
[Audio Steps \(former Voice tab\)](#)  
[Beep](#)  
[Play](#)  
[Record](#)  
[Say](#)  
[Set Play Variable](#)  
[Set Volume](#)  
[Speak Text](#)  
[Send TT](#)  
[Data/XML Steps](#)  
[DB Connect](#)  
[DB Disconnect](#)  
[DB Lock](#)  
[SQL Begin Transaction](#)  
[SQL End](#)  
[SQL End Transaction](#)  
[SQL Fetch](#)  
[SQL Statement](#)  
[XML End](#)  
[XML Fetch](#)  
[XML Load](#)  
[XML Write](#)  
[Documentation Steps \(Doc. tab\)](#)  
[Connect Point](#)  
[Notes](#)

[Save Log](#)  
[Step Comments](#)  
[Write Text](#)  
[Programming Steps \(Prog. tab\)](#)  
[Case](#)  
[Compare](#)  
[Copy File](#)  
[Create Folder](#)  
[Date Info](#)  
[Delete File](#)  
[DLL Function](#)  
[Encrypt](#)  
[Fatal Stop](#)  
[Financial Calc](#)  
[Global Event](#)  
[Go To Label](#)  
[Hash](#)  
[Label](#)  
[Load Text](#)  
[Math](#)  
[Parse](#)  
[Random Number](#)  
[Read Registry](#)  
[Reload Off](#)  
[Reload On](#)  
[Run Basic Script](#)  
[Run Flowchart](#)  
[Run Program](#)  
[Run TAP](#)  
[Save Log](#)  
[Scan Folder](#)  
[Set Global Variable](#)  
[Set Language](#)  
[Set Local Variable](#)  
[Set Play Variable](#)  
[Wait](#)  
[Write Text](#)  
[Advanced Steps \(Adv. tab\)](#)  
[Set HW Parameter](#)  
[SR](#)  
[Menu](#)  
[Speak Text](#)  
[SR Acquire](#)  
[SR Barge-In On](#)  
[SR Barge-In Off](#)  
[SR Clear Voice Buffer](#)  
[SR Flag Result](#)  
[SR Get Digits](#)  
[SR Get String](#)  
[SR Get Yes/No](#)  
[SR Release](#)  
[SR VoicePass Auth](#)  
[SR VoicePass Reg](#)  
[Web Steps](#)  
[HTTP](#)  
[Send Email](#)  
[TCP Connect](#)  
[TCP Disconnect](#)

[TCP Receive](#)  
[TCP Send](#)  
[UDP Bind](#)  
[UDP Close](#)  
[UDP Receive](#)  
[UDP Send](#)

### 5.6.1.1 START and FINISH steps

The START and FINISH steps in TeleFlow aren't steps in the same sense that others are. The TeleFlow Application developer can't add new START or FINISH steps to an application, as a single one of each exists in a flowchart as soon as it is created. Similarly, the TAP Screen has a "Starting Point" step, which is used to indicate the very first flowchart to run when starting the TeleFlow Application. The topics in this chapter describe what each of these steps is for.

[Starting Point](#)  
[START step](#)  
[FINISH step](#)

#### 5.6.1.1.1 Starting Point

Sets the first flowchart to run when the application starts, and can be set up to capture the [Application Parameters @P1 ... @P9](#) into global variables. Every TeleFlow Application has just one Starting Point step, located in the TAP Screen.

### PROPERTIES

**Parameter Value (@P1 - @P9):** OPTIONAL: The application parameter variables @P1 through @P9 can optionally be captured to global variables by populating these fields. Regardless of whether or not they are saved to new global variables, @P1 to @P9 retain the values passed into them. NOTE: This only works in conjunction with the [Run TAP](#) step. When @P1 to @P9 are set using [TeleFlow Line List](#), or with TeleFlow Simulator parameters, the optional variables are not set.

**0:Value/1:Ref:** Pass in method for the application parameter. The options are:

- 0:Value: Pass by Value
- 1:Ref: Pass by Reference

See the description on [Flowchart Parameters](#) for more information: Passing application parameter variables by value and reference work the same way for TeleFlow Applications as they do for flowcharts.

**Description:** Comments/description field for the parameter. These are displayed next the the Starting Point step.

### RELATED STEPS



[START step](#)



[FINISH step](#)

#### 5.6.1.1.2 START step

The START step is used to indicate which step to start the flowchart with, and to indicate what flowchart parameter variables the flowchart will accept, if any.

The comments section of the START step become automatic documentation for the flowchart (i.e. When a Run Flowchart is used to run it, or when viewed in the TAP screen, these comments are visible next to the step).

## PROPERTIES

**Receiving variable (1-10):** The TeleFlow variable to receive the value passed into the flowchart by the [Run Flowchart](#) step. This variable is automatically created as a [local variable](#) and populated with the value passed in.

**Pass in Method:** Pass in method for the parameter. The options are:

- **Pass by Value:** The receiving variable for any parameter passed by value is simply set to the value passed in. If nothing is supplied in the Run Flowchart step, the receiving variable is set to empty string.
- **Pass by Reference:** When a flowchart reaches the [FINISH step](#), any variable passed by reference is updated in the flowchart that passed it in. Eg. @MY\_VARIABLE contains 5, and is passed by reference into flowchart 'Add\_One' in parameter 1, which receives the variable as @NUMBER\_TO\_INCREMENT. The 'Add\_One' flowchart changes @NUMBER\_TO\_INCREMENT from 1 to 2, then the flowchart exits. Because the variable was passed by reference, @MY\_VARIABLE is updated to 2.

**Description:** Comments/description field for the parameter. These are displayed next to the the START step.

## RELATED STEPS



[Starting Point](#)



[FINISH step](#)

### 5.6.1.1.3 FINISH step

The end of the flowchart. When this step is reached, the application continues from the [Run Flowchart](#) step that ran it.

If the FINISH step in the very first flowchart run for the application (as defined in the TAP Screen with the [Starting Point](#) step) is reached, the application exits.

If a [Run TAP](#) was used to run the TeleFlow Application, application flow continues in the application that ran the TAP from just after the Run TAP step.

## PROPERTIES


None.

## RELATED STEPS



[START step](#)

### 5.6.1.2 Action Steps

Almost every step has at least one connector , or dangler, attached to it. The connector, when dragged and dropped onto another step, connects the step it is attached to to the step it is dropped on. This creates a flow line from the connector's step to the step it is dropped on. When a step has only one connector, this is its SUCCESS connector, and the flow it creates between two steps is the SUCCESS path (i.e. When the step does what it was supposed to do without error, it follows this path to the next step).

In addition to the success connector, steps may have "Action Steps" hanging off of them, and/or available to be used. Action Steps provide a way to create optional call flow when a step leaves for a reason other than successfully completing to task for which it exists (i.e. They allow you to create an alternative action to take from that of the success path). Some steps don't have a success path at all, instead having action steps related to actions taken by the caller, or based on logical decisions within the application.

Some steps don't show (all) their action steps when added to an application, because they aren't always used. The Play step, for example, has a Fail action step that can be used to set a Fail path so that an application can do something else if the file to play doesn't exist. This is optional because a missing file is normally an error the application can't do anything about.

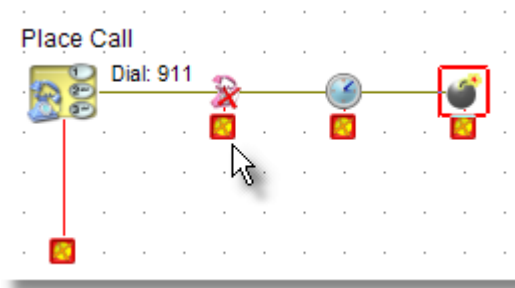
To show action steps for a step that doesn't show them by default, right-click the step in the flowchart space, and select the "Unhide Action Steps" option.

The following list is comprised of all of the action steps that may be associated with steps in TeleFlow.

- [Busy](#)
- [Connector Icon](#)
- [Failure Action](#)
- [Invalid Action](#)
- [Telephone Buttons](#)
- [Time-Out Action](#)
- [Unrecognized Voice-Action](#)
- [Yes / No Action Steps](#)

#### 5.6.1.2.1 Busy

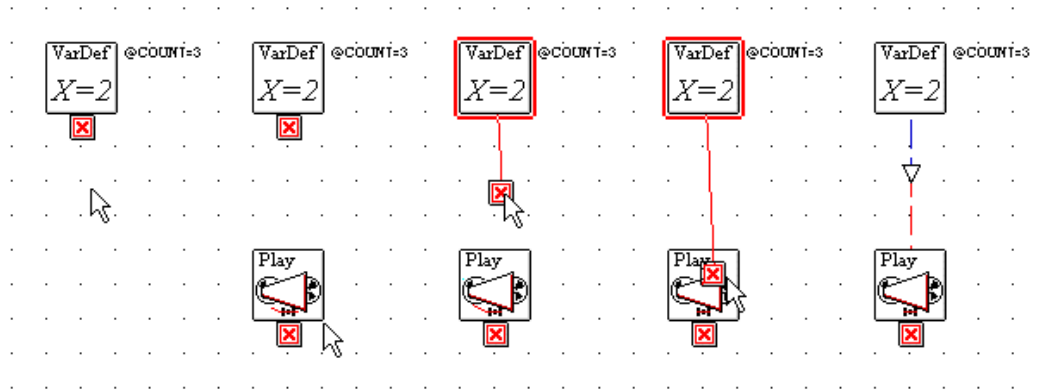
The Busy step is an Action Step that defines the path the application will follow if a BUSY result occurs when attempting to place a phone call. Only [steps](#) that can place phone calls can have a Busy Action Step. The busy step is indicated by the mouse in the picture below.



#### 5.6.1.2.2 Connector Icon

A *Connector* icon (sometimes referred to as "the dangler") is attached to almost all [Steps](#) and [Action Steps](#). *Connector* icons are used to join steps together to create flow.

Once a new step has been placed on the Flowchart, you can click and drag the connector to another Step icon. Once connected, the *Connector* icon will disappear and a line will join the two steps with an arrow indicating flow direction. This represents the flow from one step to the next. By joining every step to another step, a completed *flowchart* is developed. The following is a detailed example of how this is done.



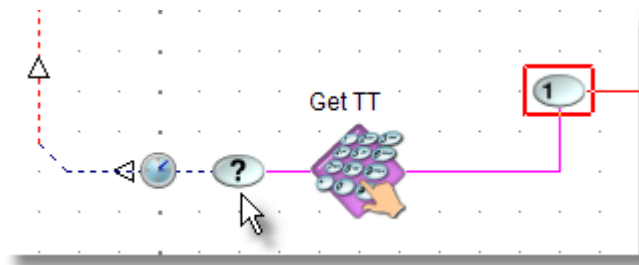
#### 5.6.1.2.3 Failure Action

The Failure step is an Action Step that defines the path the application will follow if the main step fails to perform its function. An example of a Failure would be if an SQL Fetch Step that could not fetch a record because there were no further records to fetch. Only [steps](#) that have a possibility of failing have a Failure Action Step (typically, they are steps where the application can reasonably decide what to do after a failure occurs as well). A Failure Action step is shown in the picture below. Normally the Play Step would have the Failure hidden, but it can be revealed from the properties of the Play Step.



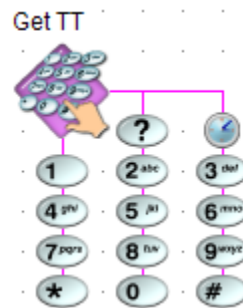
#### 5.6.1.2.4 Invalid Action

The Invalid action step is found on most step icons that accept input from the telephone (Menu, Get TT String, etc). The invalid path will be followed if the caller inputs something other than what the application is expecting. For example, assume the application has offered the caller the option of pressing one, two, or three. If they accidentally press four (which no path is programmed for), the Invalid path will be followed. Typically, this path ends in a script that says something like: "You have entered a number that was not given as an option. We will play the options again."

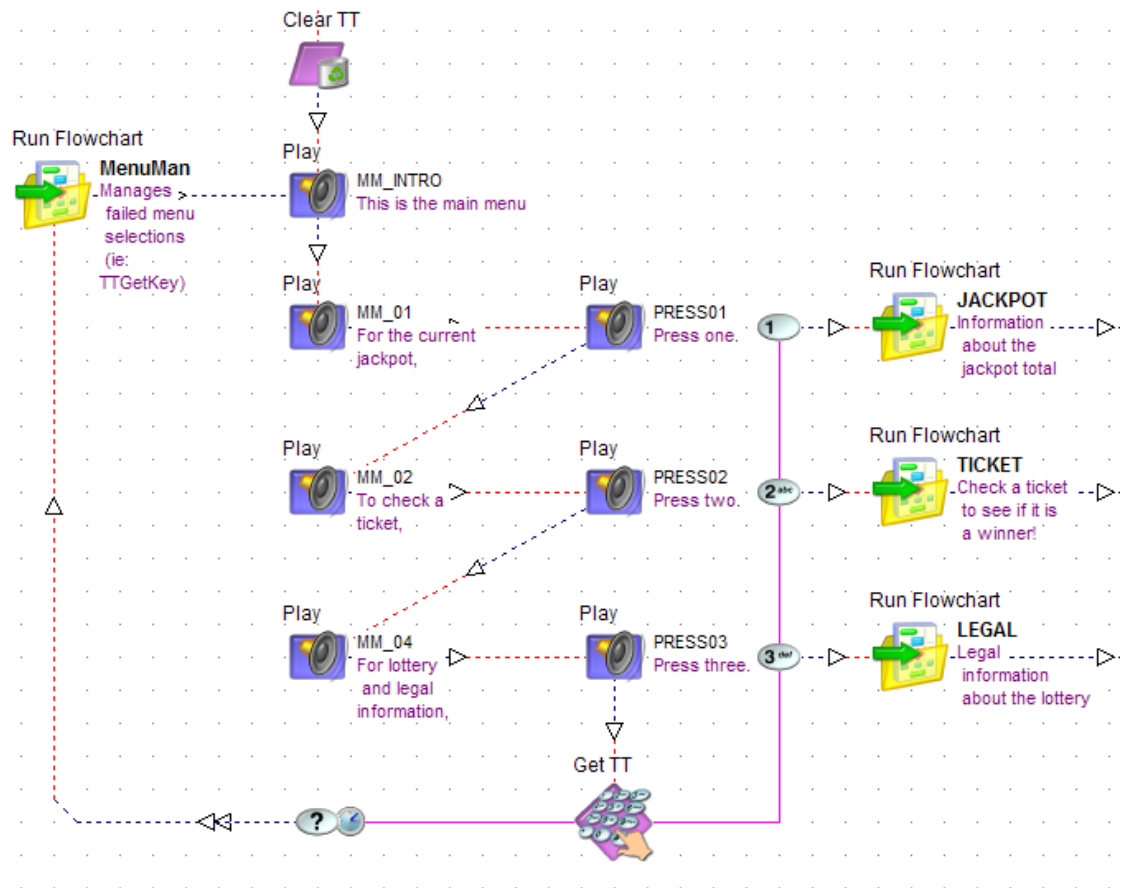


#### 5.6.1.2.5 Telephone Buttons

The telephone button icons represent the keypad found on most touch tone telephones. They include icons for the pound and asterisk keys. They are used typically when the developer wants to receive input from the caller via touch tones. If the caller presses "1", then the Digit-One Action Step path would be followed. There are other Action steps that are available from certain [Menu](#) steps, such as the letters A,B,C,D. Each action step that stems from a Menu step has its own [connector icon](#).

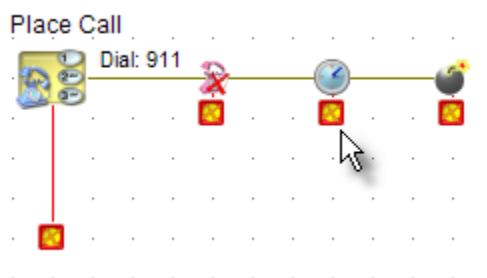


Here is an example of the Action Steps of a Menu in the demonstration application Lottery.Tap.



#### 5.6.1.2.6 Timeout Action

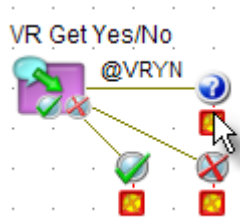
The Timeout step is an Action Step that defines the path the application will follow if a specified time limit is surpassed without any action being performed by the Step. Only [steps](#) that support a time limit have a Timeout Action Step. The Timeout step is pointed out in the picture below.



#### 5.6.1.2.7 Unrecognized Voice Action

This action step is used only with the speech recognition steps. This action step path is followed when an IVR receives an unrecognizable voice response. An Unrecognized Voice Action Step of an SR Get Yes/No Step is pointed out in the picture below.





#### 5.6.1.2.8 True / False or Yes / No Action Steps

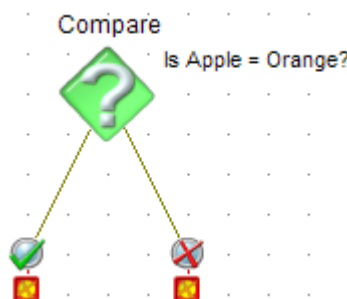
Several Step Icons make comparisons between two objects or two values. The True/False Action Steps offer flow paths depending on the outcome of the comparison.

E.g. Is variable @FILE = 2?

If the query is made, and the variable does equal 2, then the 'true' path will be followed.  
If the query is made, and the variable does not equal 2, then the 'false' path will be followed.

#### See Also:

[Compare Step](#)



#### 5.6.1.3 Phone Steps

The following list is comprised of all of the phone steps that are available in TeleFlow Designer. These steps are found under the Phone Tab of the [Steps Toolbox Window](#).

- [Block Port](#)
- [Open Conference](#)
- [Join Conference](#)
- [Leave Conference](#)
- [Close Conference](#)
- [Disable Reorder](#)
- [Enable Reorder](#)
- [Add to Fax](#)
- [Fax Clear](#)
- [Fax Receive](#)
- [Fax Send](#)
- [Flash Hook](#)
- [Global Event](#)
- [Hang Up](#)
- [Pick Up](#)
- [Place Call](#)
- [Register SIP URI](#)

[Send TT](#)  
[Set Media for VoIP](#)  
[Set Reorder Tone](#)  
[Switch Call](#)  
[Switch Disconnect](#)  
[Transfer Call](#)  
[Unblock Port](#)  
[Wait for Call](#)

#### 5.6.1.3.1 Block Port

Blocks callers from calling in. This step is used to provide a busy signal when the port/channel needs to ignore incoming calls. A block is removed with the '[Unblock Port](#)' step.

#### PROPERTIES

None (affects the current port only)

#### ACTION STEPS

None

#### SUPPORTED TELEPHONY HARDWARE PLATFORMS

NMS Communications

#### RELATED STEPS



[Unblock Port](#)

#### 5.6.1.3.2 Conference

The following topics describe the steps available for creating and managing telephone conferences.

[Open Conference](#)  
[Join Conference](#)  
[Leave Conference](#)  
[Close Conference](#)

#### 5.6.1.3.2.1 Open Conference

Creates a conference bridge. Callers can then join the bridge and have a multi-person conversation.

#### PROPERTIES

**Variable to receive conference handle:** When a conference is successfully created, TeleFlow will store an identifier for this conference in the variable you provide. This identifier is required in order to join members to the conference (see the [Join Conference step](#)) or to close the conference when it is completed (see the [Close Conference step](#)).

**Maximum number conference seats:** This setting determines the maximum number of members your conference will hold. The term "seats" draws an analogy between a conference

room and a telephone conference. You do not need to add the maximum number of members to your conference in order to begin a conference. However, you cannot add more members than you have seats defined. This is required because conferencing is resource-intensive, and conference resources must be set aside during the conference setup.

**Echo cancellation:** Echo cancellation prevents each member's audio from being echoed back to him down the "ear path" of the telecom circuit. It consumes more resources from your telephony card(s), and therefore is not recommended, unless your participants find the echo unpleasant.

**DTMF clamping:** DTMF clamping prevents a touch-tone pressed by one conference participant being heard by all participants in the conference. This is especially important if your application is programmed to use TT input to exit a member and provide him or her with other options. For example, you may wish the star key (\*) to allow a participant to leave the conference to select a different conference to join. With DTMF clamping OFF, all participants would be pushed out of the conference if just one of them pressed star.

**Tone clamping:** If your telecom circuit uses different tones than DTMF, you may need to turn on Tone clamping instead of, or in addition to DTMF clamping.

## ACTION STEPS



**Failure:** A failure can occur if telephony hardware lacks sufficient resources to start the conference with the parameters specified. This may be because the hardware lacks the capabilities for a conference, or because the resources are consumed by other conferences.

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

NMS Communications

## RELATED STEPS



[Join Conference](#)



[Leave Conference](#)



[Close Conference](#)

### 5.6.1.3.2.2 Join Conference

Joins the current port to a specified conference bridge so that the caller can participate in a multi-person conversation.

## PROPERTIES

**Conference handle:** The identifier of the conference to join. Conference handles are set when a conference is created with the [Open Conference step](#).

**Automatic gain control:** This setting will adjust the members volume if he or she is too quiet or

too loud.

**Introductory beep:** Indicates whether or not to play a short tone to all participants of the conference when the member joins or leaves the conference. The options for this setting are:

- **none:** Conference does not beep for this member
- **beep on member join only:** The conference will beep when this member joins but not when he or she leaves
- **beep on member join and exit:** The conference will beep when this member joins and when he or she leaves

**Introductory play:** Sets an audio file (or files) to play when the conference member joins. Use this option with discretion, as the audio will interrupt the conversation currently in progress in the conference. The Join Conference step supports up to two plays, each of which may be a TeleFlow script code, or an audio file.

**1st play code: or file:** Supply either a TeleFlow script code, or an audio file, but not both. If an audio file is specified, the format of the file is determined by the "Voice file type" field (see below). Leaving these fields empty means no introductory audio will be played in the conference.

**2nd play code: or file:** Supply either a TeleFlow script code, or an audio file, but not both. If an audio file is specified, the format of the file is determined by the "Voice file type" field (see below). Leaving these fields empty means no introductory audio will be played in the conference.

**Voice file type:** If playing a file or files, this specifies the file format of the audio contained in the file. If the wrong format is specified for the file(s), TeleFlow Server may fail on the Join Conference step, or strange noise may be played in the conference.

## ACTION STEPS



**Failure:** A failure can occur if the conference has not been created yet, has been closed or is full and cannot accept new members.

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

NMS Communications

## RELATED STEPS



[Open Conference](#)



[Leave Conference](#)



[Close Conference](#)

## 5.6.1.3.2.3 Leave Conference

Causes the conference member to exit the conference.

**PROPERTIES**

None

**ACTION STEPS**

None

**SUPPORTED TELEPHONY HARDWARE PLATFORMS**

NMS Communications

**RELATED STEPS**

[Open Conference](#)



[Join Conference](#)



[Close Conference](#)

## 5.6.1.3.2.4 Close Conference

Closes a conference previously created by the [Open Conference step](#).

NOTE: This step will generate an error if the conference handle is invalid. This means care must be taken with how you manage your conference close. A conference handle created before TeleFlow Server or an application instance are shut down will not be valid when the application is restarted.

**PROPERTIES**

**Conference handle:** The identifier of the conference to close. Conference handles are set when a conference is created with the [Open Conference step](#).

**ACTION STEPS**

None

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

NMS Communications

## RELATED STEPS



[Open Conference](#)



[Join Conference](#)



[Leave Conference](#)

### 5.6.1.3.3 Disable Reorder

Disables reorder tone detection on the current port. See [Enable Reorder](#) for more information.

## PROPERTIES

None (affects the current port automatically)

## ACTION STEPS

None.

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

Ai-Logix

Dialogic

### 5.6.1.3.4 Enable Reorder

Enables reorder tone detection on the current port. On analog lines (only), a caller hanging up is detected automatically based on voltage changes. In many analog systems, these voltage changes do not occur, but reorder tones are generated on the line. Using this step and enables TeleFlow to hang up when these tones are detected. The application must first use the [Set Reorder Tone](#) step to configure the tone frequencies and cadence. The frequencies and cadence are generally based on the area of the world the application is run in.

## PROPERTIES

None (affects the current port automatically)

## ACTION STEPS

None.

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

Ai-Logix  
Dialogic


### 5.6.1.3.5 Fax Steps

The following list is comprised of all fax steps in TeleFlow Designer. These can be found in the Phone tab of the [Steps Toolbox Window](#).

[Add to Fax](#)  
[Fax Clear](#)  
[Fax Receive](#)  
[Fax Send](#)

#### 5.6.1.3.5.1 Add to Fax

Adds a document to the Fax Queue to be faxed the next time a Fax Send Step is executed. Any number of documents can be added to the Fax Queue, one at a time. They will all be faxed out in the

same order they were added to the queue, when the  Fax Send step is initiated.

## PROPERTIES

**Fax image file (TIFF-F) to add to fax queue:** The directory path to and filename of the document to add to the Fax Queue. The entry for this field can be a TeleFlow Variable, provided that the variable contains a valid filename. The file to be faxed must be a Tagged Image Format (.TIF) file. Please refer to the FaxBuilder product for information on how to generate fax pages with text and graphics.

## ACTION STEPS

None.

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

NMS Communications

## RELATED STEPS



[Fax Send](#)

#### 5.6.1.3.5.2 Fax Clear

Removes all queued documents from the Fax Queue. Documents are queued with the [Add to Fax](#) step

## PROPERTIES

None.

#### **ACTION STEPS**

None.

#### **SUPPORTED TELEPHONY HARDWARE PLATFORMS**

NMS Communications

#### **RELATED STEPS**



[Fax Send](#)



[Add to Fax](#)

##### 5.6.1.3.5.3 Fax Receive

Waits for and answers an in-bound fax calls. This step assumes in-bound calls originate from fax machines, and performs the appropriate hand shaking routines.

#### **PROPERTIES**

**Fax image file (TIFF-F):** The inbound fax will be placed in the filename and directory indicated.  
e.g.:c:\ivrapp\fax\faxin.fax or @FAXFILE

#### **BUTTONS**

**Locate:** Opens the file location dialog window. Use this button to find a fax file to send.

#### **ACTION STEPS**

None.

#### **SUPPORTED TELEPHONY HARDWARE PLATFORMS**

NMS Communications

#### **RELATED STEPS**



[Fax Send](#)




## 5.6.1.3.5.4 Fax Send

Initiates a fax call. If the call is answered, it performs the appropriate fax handshaking routines, and sends a fax.

**PROPERTIES**

**Telephone number to dial:** The telephone number to place a call out to. Number must be passed in numerical format (I.e. No dashes or parentheses). e.g.:16045551234 or @FAX\_NUMBER


**Rings before TIMEOUT:** The number of rings to wait for the remote side to answer before following the path of the  ['Time Out'](#) action step. e.g.: 4 or @RINGS

**Optional fax image file (TIFF-F) to send:** The fax send step will already send any faxes that were queued prior to the call with the [Add to Fax](#) step. This field specifies an optional, single fax image file to send when the fax call is answered. This can be done either instead of sending faxes in the queue, or in addition to it. e.g.:c:\ivrapp\fax\faxin.fax or @FAXFILE


**BUTTONS**

**Locate:** Will pop up a file location dialog window. Use this button to find a fax image file.

**ACTION STEPS**

 [Time Out](#): The outgoing call was not answered in the number of rings indicated in the "Rings before TIMEOUT" property.

 [Busy](#): The remote side (called party) was busy or engaged.

 [Fail](#): The outgoing call failed. This could indicate that the line was not able to dial out, perhaps because there was no dial tone when it took the phone off the hook.

**SUPPORTED TELEPHONY HARDWARE PLATFORMS**

NMS Communications

**RELATED STEPS**

[Fax Receive](#)



[Add to Fax](#)

## 5.6.1.3.6 Flash Hook

Initiates a short term hang up with a central telephone switching system. This is equivalent to pressing and releasing the telephone receiver. Flash hooks are sometimes used in analog circuits to initiate call transfers or 3-way calls. (In the case of call transfers, it is typically easier to take advantage of the features of the [Place Call](#) step, which already has an initial flash hook built into it) The availability of such services on your telecom circuit will limit how and when the Flash Hook step can be used.

## PROPERTIES

**On-hook time (ms):** The length of time this step will flash hook for. Note: 500ms = Flash hook, but 1000ms is often used by the local telephone company for residential services.

## ACTION STEPS

None

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

Ai-Logix  
Dialogic  
NMS Communications

## RELATED STEPS




[Place Call](#)



[Send TT](#)

### 5.6.1.3.7 Global Event

Sets the [Labels](#) to go to for special events that may occur during the operation of an application. At any time during the flow of the application, events can be directed to different  [Labels](#). When one of the set events occurs, the TeleFlow Server or Simulator will search the current flowchart, then the previous flowchart (i.e. the flowchart that ran the current flowchart), and so on, essentially backtracking the direction the user came, for the first instance of the Label name defined. This is where the Server will resume the flow of the application (i.e. When the Label step is found, the application will continue from it). An important function of the Global Event step is to indicate where to continue running from when the caller hangs up. Often, logging will be placed at the Global Hang-Up label to track call duration and

other aspects of the call before returning to a  [Wait For Call](#) step to wait for the next call to come in.

When defining a global label, it is best to ensure that the matching Label step is in the same flowchart as it is being defined in. (If for no other reason than it is easier to read your application this way) It is also acceptable to set the global label to a label that is certain to exist in a flowchart higher up in the application/call flow.

See the [Go to Label](#) topic/step for more information about how labels work in TeleFlow.

## PROPERTIES

**The type of event:** Sets the type of event to define a label for, per the following options:

- **Hang-up:** When the user hangs up, go to the label name defined.
- **Fatal error:** When an error occurs that the TeleFlow Server/Simulator is not able to handle, or the application reaches a [Fatal Stop](#) step, go to the GLOBAL label name defined.

**GLOBAL label to go to for event:** The name of the label to jump to when the event set above occurs. e.g.: START

### ACTION STEPS

None.

### RELATED STEPS





#### 5.6.1.3.8 Hang Up

Hangs up the current call, and releases the telephone line.

### PROPERTIES

**Hang Up Option:** Determine if progress should continue to the next step, or go to the [GLOBAL HANG UP](#) label.

- Hang up and continue to the next step follows the .
- Hang up and go to the [GLOBAL HANG UP](#) label. ( will disappear when this option is selected)

### ACTION STEPS

None

### SUPPORTED TELEPHONY HARDWARE PLATFORMS

Ai-Logix  
Dialogic  
NMS Communications

### RELATED STEPS



#### 5.6.1.3.9 Pick Up

Takes the telephone port off-hook, or otherwise answers an in-bound call that was not answered because the preceding [Wait for Call](#) step was set to "Do not pickup after rings".

#### PROPERTIES



None. (affects the current port automatically)

#### SUPPORTED TELEPHONY HARDWARE PLATFORMS

Ai-Logix  
Dialogic  
NMS Communications

#### SPECIAL HARDWARE CONSIDERATIONS

On analog circuits, you can use the Pick Up step to take the phone off-hook. Digital circuits typically require that a call is offered by the network before the port/channel can be taken off-hook.

Not picking up the phone when a call is offered to the  [Wait for Call](#) step, and subsequently using Pick Up step, or  [Hang Up](#) step is only fully supported on digital telephone circuits. This allows the application to detect an incoming call, check if the number dialed to reach the system equates to a particular service offered on the system, and either route the caller to the appropriate service, or hang up to reject the call.

#### RELATED STEPS



[Wait for Call](#)



[Hang Up](#)

#### 5.6.1.3.10 Place Call

Initiates an outgoing telephone call. The Server can place a new call by allocating either a new telephone line, or by using the currently utilized telephone line. Most calls will probably use 'Initiate connection' unless a central telephone switching system is being used to transfer or conference calls.

No matter the result of the attempt to place an outbound call, the system variables [@CALLOUT\\_RESULT](#) and [@CALLOUT\\_REASON](#) will be set to values indicated what the result was, and why that result occurred.

#### PROPERTIES


**Telephone number to dial:** The telephone number to place a call out to. Number must be passed in numerical format (I.e. No dashes or parentheses). e.g.:16045551234 or @TELNUMBER


**Rings before TIMEOUT:** The number of rings to wait for the remote side to answer before

following the path of the  [Time Out](#) action step. e.g.: 4 or @RINGS

**Connection:** Indicates the type of connection to use to place the call. The options are:

- **Initiate connection:** Takes the current port off-hook to get a single party on the line.
- **Current connection:** Uses a flash-hook to put the current party on hold, and dials out to a second party in an attempt to transfer the call. This is only possible if flash-hook transfers are supported by the lines in use.

**Connect on CED (will connect to a Fax):** If checked, place call will connect the call if a fax (CED) tone is detected during the out-bound call. Otherwise, TeleFlow will not connect, and will follow the  [Fail](#) path.

**Connect on SIT (will remain connected after SIT tone):** If checked, place call will connect the call if a special information tone (SIT) is detected during the out-bound call. Otherwise, TeleFlow will not connect and follows the  [Fail](#) path.

**Call Progress Analysis / Answering Machine "Detection" settings:** The following three fields work in tandem to provide information to the application about the voice detection phase of call progress analysis.


- **Human Variable to receive initial voice length:** Receives the duration of the voice that was detected answering the phone. The value received will be measured in milliseconds. This field is optional, and the default is blank.
- **Silence TIMEOUT:** Specify a value in milliseconds in the Silence timeout(ms) field to override the default end-of-voice detection. The silence timeout is the minimum duration of silence following the end of the initial voice that will cause call progress analysis to end with a result of voice-detected.
- **Maximum voice length (ms):** In the Maximum voice length (ms) field, specify the maximum time to wait for the voice to end before exiting voice length detection. The default value supplied by the Designer is 60000 ms (1 minute). This value ensures the voice detection will end, even if it encounters answering audio that lasts an extremely long time.

**Caller Id to send (ISDN or SIP only):** If the line protocol in use supports it, and the Telco will send it, this field indicates the caller id to send to the remote party (i.e. This is what a remote party will see as caller id when they receive a call initiated by the Place Call step). Note that this setting carries no guarantees: There are situations that cause the caller id sent by your application to not make it all the way to the remote side. This is particularly likely when crossing country borders, but other boundary issues may exist as well.

**ISDN Calling number plan:** Use this field to set the ISDN Number Plan of the calling number (ie: the phone number associated with the TeleFlow Port placing the call). This feature is only available on ISDN circuits. If unsure, leave setting at "DEFAULT".

**ISDN Calling number type:** Use this field to set the ISDN Number Type of the calling number (ie: the phone number associated with the TeleFlow Port placing the call). This feature is only available on ISDN circuits. If unsure, leave setting at "DEFAULT".

## ACTION STEPS

 [Time Out](#): The outgoing call was not answered in the number of rings indicated in the "Rings before TIMEOUT" property.

 [Busy](#): The remote side (called party) was busy or engaged.



**Fail:** The outgoing call failed. This could indicate that the channel was not able to dial out, perhaps because there was no dial tone, or a bad phone number was supplied.

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

Ai-Logix  
Dialogic  
NMS Communications

## SPECIAL HARDWARE CONSIDERATIONS

Using "current connection" to place a second call will only work if the telephone circuit the port is connected to supports call transfer.

**Ai-Logix:** As of TeleFlow 2006, Place Call is now supported on Ai-Logix active hardware. However, the ability to place a second call using "current connection" is still under development for Ai-Logix hardware.

**NMS Communications:** On NMS hardware, the flash-hook duration for using "current connection" must be set to 500 milliseconds, or left empty to use the default value (also 500 ms).

## RELATED STEPS



[Wait For Call](#)



[Flash Hook](#)

### 5.6.1.3.11 Register SIP URI

Registers a SIP URI with a SIP registrar.

## PROPERTIES

**Registrar:** Host name or IP address of the SIP registrar. The host name or IP address may include a port specifier, otherwise the default SIP port, 5060, is used.

**URI to Register (To):** The SIP URI to register with the registrar. e.g. sip:Sales@@enGenic.com

**Registrant URI (From):** SIP URI of the registrant registering with the registrar. e.g. sip:VolPCO@@enGenic.com

**Contact URI:** SIP contact URI for registree. e.g. sip:JohnDoe@@enGenic.com

**Expiration Time (seconds):** SIP registrations have an expiration timeout in seconds. This field is populated by default with 1 hour (3600 seconds).

**Variable to Load Registration Response Header into:** TeleFlow variable to load with the header of the SIP REGISTER response, or any socket error messages.

**Response Timeout (ms):** Maximum time to wait, in ms, for a response before following the TIMEOUT path. If this field is left empty, @TIMEOUT (3000 by default) is used.

## ACTION STEPS



## RELATED STEPS



### 5.6.1.3.12 Send TT

Generates touch-tones. This can be used to automate a call or to communicate with other telephone services.

## PROPERTIES

**Touch tones to play:** The touch tones to generate. Note the following:

- May include: '1 2 3 4 5 6 7 8 9 0 A B C D \* #' e.g.: 123 or @TONES
- Variables should be populated with characters from the above list only.
- Characters not in the list above will be ignored/skipped.

## ACTION STEPS

None

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

Ai-Logix  
Dialogic  
NMS Communications

## RELATED STEPS



### 5.6.1.3.13 Set Media for VoIP

Sets parameters required for SIP VoIP support in an application.

## PROPERTIES

**Media IP Address:** The IP address the VoIP media transmits across. On systems with multiple network connections it is important to set the correct IP address. If the network will dynamically re-assign this value, some external mechanism must provide that information back to the application so that these settings can be updated.

**Media IP Port:** The port for VoIP media transmission for this application. Generally speaking, it is a good idea to have a variable @VOIP\_PORT that is set to some base value, plus twice the application port number. Example: assign @VOIP\_PORT to 7999 + @PORT

**Media Payloads:** This is a prioritized list of the media payloads that are acceptable for this application. Example: for SIP, "0 8 4" indicates that the preferred payload is G7.11 uLaw, followed by G7.11 aLaw, followed by G7.26.

The numeric codes used follow the Payload Type Definitions in RFC 3551. The most common media payloads supported by TeleFlow are in the following table:

Payload Type	Encoding
0	G711_U
3	GSM
4	G723
8	G711_A
18	G729

**Prioritize Client/Server Payloads:** When a SIP call comes in, the standard specifies that the highest priority client media payload that is also in the server's media payloads list, is the payload that should be used (default behaviour: Client). This may be undesirable for bandwidth reasons. Selecting the Server radio button chooses the highest priority (first in list) server media payload that is also in the client's media payloads list.

## ACTION STEPS

None.

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

NMS Communications

## RELATED STEPS



[Register SIP URI](#)



[Wait For Call](#)

### 5.6.1.3.14 Set Reorder Tone

Sets reorder tone cadence and frequencies, and enables reorder-tone detection.

Different parts of the world have different reorder-tone standards, and even telephony equipment within the same part of the world may use different standards. Most sound like a busy signal at double-speed(they are sometimes referred to as "fast-busy"). This step is populated by default with the



North American standard:

Frequency 1	480 Hz
Frequency 2	620 Hz
Minimum On Time	200 ms
Maximum On Time	300 ms
Minimum Off Time	200 ms
Maximum Off Time	300 ms

Please consult your CO or PBX manual to determine what reorder tone settings are appropriate for your system.

On analog lines on supported telephony hardware, reorder-tone detection is on by default with the North American standard settings.

## PROPERTIES

**Frequency 1:** Specify the first (or only) frequency in this field. Most reorder tones consist of 2 tones.

**Frequency 2:** Specify the second (or only) frequency in this field. Most reorder tones consist of 2 tones.

**Minimum On Time:** This is the minimum required time for the frequencies to be heard in the reorder-tone cadence.

**Maximum On Time:** This is the maximum allowed time for the frequencies to be heard in the reorder-tone cadence.

**Minimum Off Time:** This is the minimum required time for the frequencies to be silent in the reorder-tone cadence.

**Maximum Off Time:** This is the maximum allowed time for the frequencies to be silent in the reorder-tone cadence.

## ACTION STEPS

None.

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

Ai-Logix  
Dialogic

## RELATED STEPS



[Enable Reorder](#)



[Disable Reorder](#)

## 5.6.1.3.15 Switch Call

Connects two calls together.

**PROPERTIES**

**Connect current port to port number:** Sets which port to connect to. e.g.: 9 or @OTHER\_PORT

**ACTION STEPS**

None.

**SUPPORTED TELEPHONY HARDWARE PLATFORMS**

Ai-Logix  
Dialogic  
NMS Communications

**SPECIAL HARDWARE CONSIDERATIONS**

In order to use the switch step, the telephony card you use must also support switching.

If you wish to connect two ports together that are on different cards, the cards must support MVIP or H.100 and have the appropriate cable connecting them.

**RELATED STEPS**

[Switch Disconnect](#)

## 5.6.1.3.16 Switch Disconnect

Disconnects two calls previously switched together with the [Switch Call](#) step.

**PROPERTIES**

None.

**ACTION STEPS**

None.

**SUPPORTED TELEPHONY HARDWARE PLATFORMS**

Ai-Logix

Dialogic  
NMS Communications

## RELATED STEPS



[Switch Call](#)

### 5.6.1.3.17 Transfer Call

Connects two calls together, using transfer capability of your telecommunications circuit.

At present this step only works with ISDN circuits that are configured to provide the 2B-Channel Transfer capability (sometimes referred to as "release link transfer").

## PROPERTIES

**Other port's call reference number:** In order to connect two ISDN calls together, you need to use the call reference number for each call, as provided by your ISDN circuit. TeleFlow knows the call reference number for the current port, however, you need to supply the reference number for the other port.

To obtain the call reference number, capture the value from the [@SYS\\_CALLREF](#) system variable.

## ACTION STEPS

None.

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

NMS Communications

## SPECIAL HARDWARE CONSIDERATIONS

In order to use the Transfer Call step, your application must be connected to a 2B-Transfer capable ISDN circuit.

## RELATED STEPS



[Switch Call](#)



[Place Call](#)

## 5.6.1.3.18 Unblock Port

Unlocks a port that was previously blocked by the [Block Port](#) step.

**PROPERTIES**

None

**ACTION STEPS**

None

**SUPPORTED TELEPHONY HARDWARE PLATFORMS**

NMS Communications



**RELATED STEPS**

[Block Port](#)

## 5.6.1.3.19 Wait for Call

Waits for and answers in-bound telephone calls.


**PROPERTIES**

**TIMEOUT (seconds) (0 = wait forever):** The length of time, in seconds, this step will wait for an inbound call. If in this time there is no call, the application will follow the path of the [Time-out](#)  icon. E.g.: 60 evaluates to 60 seconds, or 1 minute. If set to empty (the default) or zero the step will wait for a call indefinitely. By default, the [Time-out](#)  action step is hidden, as this feature is rarely used.

**Wait for Rings:** Indicates the number of rings to wait before moving to the next step. How this is handled is determined by the following two options:

- **Pickup after rings:** The number of rings the application will wait before picking up an incoming call. E.g.: 2 or @RINGS (application-defined variable). The default value is 2. If left empty, or set to 0, TeleFlow will pick up the call immediately. If you are connecting to an analog telephone circuit, and you wish to capture Caller ID, make sure that you set the ring count to 2 or more.
- **Do not pickup after rings:** After this number of rings, the application successfully leaves the Wait for Call step, but doesn't pickup or answer the phone. This can be used to decide whether or not to actually accept the call. (If, for example, the caller id is from a number you

have decided to block calls from, you can  [Hang Up](#) instead, and deny the call. If you

choose to pick up the phone, use the  [Pick Up](#) step.)

**Variable to receive Caller Id:** Variable to capture the telephone number that initiated the incoming call. e.g.: @CALLERINFO (application-defined variable) May be left blank.

**Variable to receive Called Id:** Variable to capture the telephone number that was dialed in order to initiate the incoming call. e.g.: @CALLERDIALED (application-defined variable) May be left blank.

**Connection delay (ms):** Because some analog circuits take time to establish a connection with the caller after the call is picked up, a connection delay is supported that will pause the application for a specified duration, in milliseconds. The default connection delay is 500. Set this field to empty or 0 to disable the connection delay.

**Protocol Specific Settings:** This section contains options that apply to specific telephony protocols only. Current options available are:

**Number of inbound digits:** For CAS/Wink Start protocol, this determines the number of incoming DNIS/Called Id digits to wait for before leaving the step.

## ACTION STEPS



[Time-out](#): No incoming call was detected in the time set in the TIMEOUT field.

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

Ai-Logix  
Dialogic  
NMS Communications

## SPECIAL HARDWARE CONSIDERATIONS

Not picking up the phone and subsequently using the Pick Up step, or Hang Up step is only fully supported on digital telephone circuits, and currently only on the NMS Communications platform.

## RELATED STEPS



[Place Call](#)



[Hang Up](#)



[Pick Up](#)

### 5.6.1.4 Input Steps

The following list is comprised of all of the Input steps that are available in TeleFlow Designer. These steps are found under the Input Tab of the [Steps Toolbox Window](#).

[Clear TT](#)  
[Get Letter](#)  
[Get TT String](#)  
[Menu](#)  
[Pending TT](#)

## 5.6.1.4.1 Clear TT

This step clears all pending touch-tones from the [touch-tone buffer](#). This is sometimes used just prior to playing a main system menu, so that callers can press a key when the system answers the phone, skip introductory messages, but still hear all options in the menu, rather than accidentally selecting one. Note that care should be taken when placing Clear TT steps in applications. Experts of a system (I.e. Frequent callers that know how the system call flow works) may expect to be able to enter options before hearing them, and this step can interfere with this.

**PROPERTIES**







None

**ACTION STEPS**

None

**RELATED STEPS**

## 5.6.1.4.2 Get Letter

This step allows callers to use their telephone keypads to enter a letter instead of a digit. Each time a button is pressed, the Server will speak back one of the letters on the telephone keypad. For example, if the caller presses 2, the Server will say 'A'. If the caller presses 2 again, the Server will say 'B'. This process continues until the caller hears the desired letter, and presses one of the exit keys,  or . The Action Steps for  or  can be used to route the application depending on how the user terminated the input. Their exact use depends on the needs of your application. One possible use of the Get Letter step is to enable the caller to spell a word using a loop. One of the action steps, perhaps the , could be used to accept the current letter and loop back to the Get Letter step to acquire the next letter. The other action step () could be used to indicate the end of the word.

**PROPERTIES**

**Variable to receive letter:** Specifies the TeleFlow Variable to store the user's selected letter. e.g.: @PENDINGTT

**TIMEOUT (ms):** The length of time - expressed in milliseconds - for which this step will wait for a touch-tone before following the path of the Time Out action step. Defaults to system variable [@TIMEOUT](#) if left blank. (e.g.: 1000 or @MY\_TIMEOUT)

**Entry Convention:** Because the telephone keypad is inconveniently missing the letter 'Q' and the letter 'Z', the entry convention field specifies which keys you want to use for them. (Remember to explain this to your caller when asking them to enter letters using this step) Your options are:

- Z and Q on the '0' button
- Z and Q on the 'Q' and 'Z' on the seven and nine key respectively

**Allowable Input:** The Get Letter Step can be used to get digits, in a similar manner to the Menu steps. This field provides the following options:






- Alphabet only
- Numbers only
- Both

Alphabet only is the manner in which you will most likely want to use this step. Numbers only



causes this function to behave very much like the [Menu Step](#), in which case, you might as well just use the Menu step. 'Both' is handy for dealing with mixed input, such as Canadian postal codes or mixed serial numbers. When Both is selected, the caller will cycle through the characters letters first, and then the number. e.g.: A,B,C,3

## ACTION STEPS

-  Star/Asterisk Touch-Tone: Indicates the path to follow if the  touch tone was pressed.
-  Pound/Number Touch-Tone: Indicates the path to follow if the  touch tone was pressed.
-  [Time Out](#): If the caller does not select a letter press a touch tone before time runs out (see TIMEOUT above), this path will be followed.

## RELATED STEPS



[Menu](#)



[Get TT String](#)



[Play](#)




[Clear TT](#)

### 5.6.1.4.3 Get TT String


Collects one or more touch-tones into a TeleFlow variable. The touch-tones collected can be entered when the application reaches this step, or can be pulled from the [touch-tone buffer](#), or both. This function is most often used to receive a string of numbers and/or symbols such as personal

identification number, account numbers, or telephone numbers.

## PROPERTIES

**Variable to receive touch-tone(s):** Any touch-tones that are entered before the input  [Time Out](#), a maximum number of touch-tones is entered, or a terminating touch-tone is pressed, will be stored in this variable. The timer on this step is reset every time a touch-tone is entered. Any touch-tone gathered into a variable by this step is removed from the [touch-tone buffer](#). e.g.: @ACCOUNTNO


**Maximum touch-tones to receive:** Defines the maximum number of touch tones that will be accepted before continuing down the path. e.g.: 6 or @MAXTOUCH

**TIMEOUT (ms):** The length of time for which this step will wait for a touch tone. If in this time one is not received, the application will follow the path of the  [Time Out](#) action step. e.g.: 3000 or @TIMEOUT:

**Allowable touch-tone(s):** Restricts touch-tones that will be accepted. Acceptable values are: 'X,A,B,C,D,#,\* ,0,1,2,3,4,5,6,7,8 or 9'. An entry of 'X' means: accept all other touch-tones that are not entered in the touch tones to terminate entry field. e.g.: 1234567890. Touch-tones not in the list are ignored (and removed from the touch-tone buffer) when encountered by this step.

**Touch-tone(s) to terminate entry:** Restricts touch tones that terminate touch tone input. Acceptable values are: 'X,A,B,C,D,#,\* ,0,1,2,3,4,5,6,7,8 or 9'. An entry of 'X' means: accept all other touch-tones that are not entered in the Allowable touch tone(s) field. Note that a touch-tone in this field and the allowable touch-tones field will terminate the entry and be accepted as input. e.g.: \*#

## ACTION STEPS

 [Time Out](#): If no touch-tones are entered before time runs out (see TIMEOUT above), this path will be followed. As long as at least 1 allowable touch-tone (see Allowable touch-tone(s), above) is entered before the time runs out, the application will follow the success path. The timer on this step is reset every time a touch-tone is entered.

## RELATED STEPS








### 5.6.1.4.4 Menu

Collects a selection through a touch tone or speech recognition, following the path related to the user's selection.




Speech recognition is only available if an [SR Acquire](#) step was previously executed. Without SR, this step will accept Touch-Tone input only.

When this step is placed on the flowchart, most of its related Action steps will also be created. Any Action steps that are not required can be hidden through an option on the right mouse button menu, or by pressing Ctrl-H on each. If the user should enter an option that is hidden, the Server will follow the path of the  All Others Action step. The four Action steps for the letters,    and , are automatically hidden. To use any of these four action steps, you must unhide them from the step's right mouse button menu.

## PROPERTIES

**Variable to receive selection:** The touch tone digit or speech recognition result will be sent to this TeleFlow variable. This field is blank by default, but may be any TeleFlow variable. Note that no result variable needs to be collected by this step; collecting a result variable with this step is optional.

**TIMEOUT (ms):** This field is the maximum time, in milliseconds, before this step will time out and follow  [Time Out](#) path if no touch tone or speech entry is detected. If this field is blank (the default), [@TIMEOUT](#) will be used.

**Speech Recognition Options:** These options are only necessary/used when speech recognition is enabled. If they are all left blank in a speech recognition enabled system, the step reverts to using touch-tone only behavior.

**THRESHOLD confidence %:** The result confidence is compared against this minimum confidence to determine a valid result. This field is populated with 50 by default. This field must not be blank.

**Variable to receive confidence %:** The result confidence will be put in this TeleFlow variable. This field is populated with [@SR\\_CONFIDENCE](#) by default. This field can be blank.

**Remember this recognition for verification through SpeechTrainer:** If the [TeleFlow SpeechTrainer](#) is active, checking this box ensures the recognition will be available for review in SpeechTrainer.

**Variable to receive SR Handle:** Use this to capture the SR Handle for the utterance if the application uses the [SR Flag Result](#) step to flag certain types of SR results in SpeechTrainer. Utterances flagged in SpeechTrainer can be found more easily, and used to isolate specific problems the application is set up to identify.

**Retain audio on THRESHOLD:** Check this checkbox to compare the speech recognition audio against another grammar (such as SR Get Yes/No or SR Get String) when the result confidence is less than the THRESHOLD confidence.

**Base grammar (name or file):** This field specifies the name of the Nuance pre-compiled grammar, or the SRGS grammar file for use with LumenVox. If this field is left blank, only the additional grammar text will be used for speech recognition. If both this field and the additional grammar field are left blank, this step will respond to touch tones only.

**Additional grammar text:** This field specifies SRGS grammar rules in addition to the base grammar, but can only be used by LumenVox. If this field is left blank, only the base grammar will be used for speech recognition. If both this field and the additional grammar field are left

blank, this step will respond to touch tones only.

### ACTION STEPS

**0** through **9**, **\***, **#**, **A** through **D**: A touch tone was detected, and the confidence was set to 100, **or** the speech recognition grammar returned a result matching the number indicated with adequate confidence. (E.g. The **0** path would be followed if a caller presses "0", or if the result from Speech Recognition was "0" and the confidence was equal to or higher than the THRESHOLD confidence %)

**?** **All Others**: A touch tone was detected that did not correspond to a connected action step, **or** a speech recognition result was detected with adequate confidence that did not correspond to a connected step. Use the variable to receive selection to determine the result.

**?** **Unrecognized Voice Action**: The confidence was less than the THRESHOLD confidence.

**?** **Time Out**: No speech or touch tones were detected before the No-Voice TIMEOUT elapsed.

### SPECIAL HARDWARE CONSIDERATIONS

- Dialogic: Only touch-tone and prompted SR are supported (I.e. SR with barge-in is not currently supported).
- Ai-Logix: Touch-tone support only.

### RELATED STEPS



[Get TT String](#)



[Get Letter](#)



[SR Acquire](#)



[SR Get Yes/No](#)



[SR Get String](#)



[SR Get Digits](#)

#### 5.6.1.4.5 Pending TT

Captures all touch-tones currently in the [touch-tone buffer](#) to a variable. The touch-tones captured by this stop are NOT removed from the touch-tone buffer, unlike other touch-tone processing steps.

To clear the touch-tone buffer or remove some keys from it, see the  [Clear TT](#),  [Menu](#),

or  [Get TT String](#) steps.

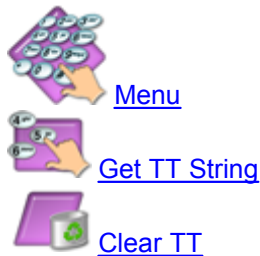
## PROPERTIES

**Variable to receive pending touch tones:** Variable to populate with all tones currently in the [touch-tone buffer](#). e.g.: @PENDINGTT

## ACTION STEPS

None

## RELATED STEPS



### 5.6.1.5 Audio Steps (former Voice tab)

The following list is comprised of all of the Audio related steps that are available in TeleFlow Designer. These steps are found under the Audio tab of the [Steps Toolbox Window](#). (Previously, this was the "Voice" tab)

[Beep](#)  
[Play](#)  
[Record](#)  
[Say](#)  
[Set Play Variable](#)  
[Set Volume](#)  
[Speak Text](#)  
[Send TT](#)

#### 5.6.1.5.1 Beep

Plays a short beep tone to the caller.

## PROPERTIES

**Interruptible:** If checked, this step will be skipped (I.e. no tone will be played) if there are pending tones in the [touch-tone buffer](#), or interrupted if a key is pressed during the beep. If the caller speaks and the system has barge-in enabled for Speech Recognition, this will also skip/interrupt the beep. If it is not interruptible, the user cannot interrupt or skip the beep.

## ACTION STEPS

None

## RELATED STEPS



[Play](#)



[Record](#)

### 5.6.1.5.2 Play

Plays an audio file to the caller.

## PROPERTIES

**Script code:** If the voice message is defined and has a code in the [Voice Script Manager](#), that code can be entered in this field. Any documentation that was done in the Voice Script Manager will duplicate itself in the notes for the Play.

**Interruptible:** If checked, this step will be skipped (i.e. no tone will be played) if there are pending tones in the [touch-tone buffer](#), or interrupted if a key is pressed during the audio play. If the caller speaks and the system has barge-in enabled for Speech Recognition, this will also skip/interrupt the Play. If it is not interruptible, the user cannot interrupt or skip the Play.

**Voice file:** Directory path to and filename of the voice file to play. If nothing is entered in the Script code field, the Server will then look in this field to find the audio file to play. E.g.:  
c:\ivrapp\vox\message.vox or @VOXFILE

**Voice type:** If playing a file from the "Voice file" entry instead of using a Script code, you must specify the type/format of the audio file. If you specify the wrong format, the play step may fail, or the audio played may sound garbled. If "By file extension" is selected, TeleFlow attempts to play the file according to the file extension and the audio formats it supports.

**Wav file options:** The options in this section are available only for WAV files, and then only for those of the TeleFlow standard Wav file format. (8kHz, 16-bit, Mono, Windows PCM Wav files)

**Message Review Mode:** If checked, the caller will be able to review the message using the telephone buttons in place of being able to skip the voice message. This option is used almost exclusively for listening to voice mail.



Rewind 2 seconds



Forward 2 seconds



Interrupts voice message play

**Play from position (ms) ... :** Plays the file from the position indicated, in milliseconds. (Eg. If set to 1500, the file starts playing from 1.5 seconds into the file.)

**For (ms) ... :** Plays the file for up to the number of milliseconds indicated. If the number is greater than the total remaining length of the file, the entire file is played (or, the file is played from the position indicated by "Play from position (ms)" to the end, if that field is being used as well).


## BUTTONS

**Voice Manager:** A short cut to the [Voice Script Manager](#). The Voice Script Manager is also accessible from the "Window" menu of TeleFlow Designer.

**Play:** Plays the voice file associated with the Script code, if the file is accessible in design time.

**Locate:** Opens a file location dialog window, which can be used to locate file to use in the Voice file entry.

## ACTION STEPS

 **Fail:** A Fail occurs on Play when the audio cannot be played. This may occur for a number of reasons, but the most likely causes are that the voice message file was not located, it was opened exclusively by another process, or the file format was not compatible with TeleFlow. Invalid file names will also cause the process to follow the Fail path. This step is hidden by default, which means that a Play that Fails normally results in a [Fatal Error](#).

## RELATED STEPS



[Say](#)



[Record](#)



[Beep](#)



[Set Volume](#)

### 5.6.1.5.3 Record

Records an audio file. Recordings finish when the caller presses a touch-tone to terminate the recording, when a set amount of silence is detected, or when the maximum recordable time limit is reached. The most common use for this function is to create voice mail, or record caller responses to questions.

## PROPERTIES

**Voice file:** Directory path to and filename of the audio file to record to. E.g.:  
C:\IVRApp\Vmail\message.wav or @VOXFILE

**Voice type:** Indicates the audio file format to save the recording to. Consult the following table for additional information about the types.

"Voice type"	File size / minute	Telephony cards supported
WAV (8kHz 16 bit PCM Mono)	960 kb/min	Usable for all cards except Dialogic
VOX (Dialogic)	240 kb/min	Only format available on TeleFlow's Dialogic implementation

VCE NMS - 24kps	180 kb/min	NMS compressed format
GSM 6.10 (13 kps)	98 kb/min	Ai-Logix only
By File Extension	as above	Records to the format indicated by the file extension, such as .WAV or .VCE.


**Do not play start recording tone (beep):** If checked, this indicates the record step should not play a tone/beep just prior to recording the audio file.


**Interruptible:** If checked, the caller can interrupt a voice message recording by pressing any telephone button. If not checked, the recording time will last for the full duration defined in maximum seconds recordable.

**Maximum time recordable (ms):** Defines the maximum recording length (in seconds) of the voice message. A setting of '0' means there is no time limit. When the maximum time is reached, the record step follows the success path. e.g.: 60000 or @MAXTIME

**Silence TIMEOUT (ms):** The maximum silence time (i.e. Time there is no sound to record, usually because the caller stopped talking) the step will wait before ending the recording. The [@RECORD\\_RESULT](#) system variable provides codes that indicate the way in which the recording completed.

## ACTION STEPS

 **Fail:** The message record failed. This may occur for a number of reasons, but the most likely cause is that the voice message file was already opened by another process. Invalid file names will also cause the process to follow this path.

 **Time Out:** If the caller does not speak before time runs out (see Silence TIMEOUT above), this path will be followed.

## RELATED STEPS



[Play](#)



[Copy File](#)



[Beep](#)



[Set Volume](#)

### 5.6.1.5.4 Say

Speaks a formatted value in a common language phrase format. This function requires that the appropriate language DLL is set up with the [Set Language](#) step, and that the prompts directory is populated with the appropriate prompt files. This directory contains all the voice messages required to properly speak any date, number or time value. See the [Set Language](#) step topic for more information on how the language settings and prompts directories work.

## PROPERTIES

**Format to speak:** Defines the way in which the data should be spoken:

- Digits: Says the digits individually, one at a time. (Eg. "111" would be "one", "one", "one")
- Number: Says the digits as a number. (Eg. "111" would be "one-hundred eleven")
- Date (yyyy/mm/dd): Says the month, day, and year portions of the supplied date. (Eg. "2007/09/13" would be "September thirteenth, two-thousand seven")
- Date (yyyy/mm): Says the month and year portions of the supplied date. (Eg. "2007/09/13" would be "September, two-thousand seven")
- Date (mm/dd): Says the month and day portions of the supplied date. (Eg. "2007/09/13" would be "September thirteenth, two-thousand seven")
- Time (12 hour): Says the time using the standard 12-hour clock, with "am" or "pm". If the seconds portion of the time is included in the "Value to speak", the step plays like this: "Twelve-thirty and five seconds pm". Providing only the hour and minute portions plays the time with the seconds. (Eg. "12:30" would be "twelve-thirty pm")
- Time (24 hour): Says the time using the 24-hour ("military") clock. If the seconds portion of the time is included in the "Value to speak", the step plays like this: "Fifteen-thirty and five seconds". Providing only the hour and minute portions plays the time with the seconds. (Eg. "15:30" would be "fifteen-thirty")
- Money (nnnnn.nn): Says a numeric value as it would normally be spoken as currency. If the amount to play has .00 for cents, the step still says "zero cents". To only play the dollar portion, don't provide the decimal portion to the "Value to speak". Likewise, if only playing the cents portion, don't include anything before the decimal point. (Eg. "13.5" would be "thirteen dollars and fifty cents")
- Alpha-numeric: Says the letters and numbers one at a time. ("V3V7J2" would be "V", "three", "V", "seven", "J", "two")

**Value to speak:** The value to be spoken in the above format. The contents of the value must conform to how it is to be spoken, per the appropriate TeleFlow variable format (Eg. Dates must be represented in YYYY/MM/DD format). Supplying data in an invalid format may cause the step to say the data in an invalid or unusual manner, or will possibly result in an error. E.g.: 1996/01/24 or @TODAYDT

**Interruptible:** If checked, the say step can be interrupted by a touch tone (or by the caller speaking if barge-in is enabled). If it is not interruptible, the caller will have to listen to the entire spoken formatted value.

## ACTION STEPS

None.

## RELATED STEPS



[Play](#)



[Beep](#)



[Set Volume](#)



[Set Language](#)

#### 5.6.1.5.5 Set Play Variable

Defines or sets a variable to a Script code (see the [Voice Script Manager](#) for more information on Script codes). The intention of this step is enhance the readability of a TeleFlow application. While you could set a variable to a Script code normally, and then use a Play step to Play the variable, doing so with Set Play Var is preferable. This is because the Set Play Var step allows you to see the script for the voice message code where you set the Script code variable. See the [Variable Usage](#) section for more information on variables.

#### PROPERTIES

**Variable to define:** The TeleFlow variable to populate with the Script code.

**Script code:** The Script code assigned to the Variable to define. Any documentation that was done in the Voice Script Manager will duplicate itself in the notes for the Set Play Variable step. E.g.: MAIN\_MENU01 or @VOICECD

#### BUTTONS

**Voice Manager:** A short cut to the [Voice Script Manager](#). The Voice Script Manager is also accessible from the "Window" menu of TeleFlow Designer.

**Play:** Plays the voice file associated with the Script code, if the file is accessible in design time.

#### ACTION STEPS

None.

#### RELATED STEPS



[Play](#)



[Set Global Variable](#)

#### 5.6.1.5.6 Set Volume

Sets the volume all voice messages are played and recorded at. Increasing or decreasing audio played on the system in this way does so relative to the file's existing volume (so, setting the volume higher system-wide could cause louder files to become too loud for your callers, although it is always best to keep the volume of most system recordings at a consistent level).

NOTE: This step sets the System Variable [@VOLUME](#). To "soft-code" volume in applications, use the [Set Global Variable](#) step to set the @VOLUME variable with a value retrieved from an external source, such as the registry or a database. To have different volumes for audio play and recorded audio, update @VOLUME just prior to any [Record](#) step, and put set it back to the application's play volume just after the step.

#### PROPERTIES

**Volume of voice messages:** The audio volume can be set to any of the following:

- -3 Quietest



- -2
- -1
- 0 Normal: This is the default, regardless of whether or not this step is used in an application.
- 1
- 2
- 3 Loudest

### ACTION STEPS

None.

### RELATED STEPS



[Play](#)



[Record](#)

#### 5.6.1.5.7 Speak Text

This step converts text to speech, playing the resulting audio to the caller.

Note: Text-to-Speech is not available as a part of the standard TeleFlow Suite. You must purchase additional software to use this step.

### PROPERTIES

**Server IP address for TTS Server computer:** The IP address of the computer hosting the TTS Server. This must be an IP address and not a computer name or URL. This field must be filled in even if the local computer is the one hosting the TTS Server.

**IP port:** IP port for the TTS Server. 7000 is the default.

**Text to speak:** Supply the text the step must convert and speak. The text can be a TeleFlow variable, or a combination of literal text and variables. E.g. "Hello @FIRST\_NAME, this is @COMPANY\_NAME calling."

**Interruptible:** If checked, this step will be skipped (I.e. no tone will be played) if there are pending tones in the [touch-tone buffer](#), or interrupted if a key is pressed while the step is playing audio. If the caller speaks and the system has barge-in enabled for Speech Recognition, this will also skip/interrupt the audio. If it is not interruptible, the user cannot interrupt or skip the step.

### ACTION STEPS



**Failure:** A failure will occur if the TTS Server does not respond, which can happen for a number of reasons. The three most common are the IP address supplied is incorrect, the IP port supplied is incorrect, or a TTS Server is not running on the computer specified by the IP address.

### SUPPORTED TELEPHONY HARDWARE PLATFORMS

Ai-Logix

Dialogic  
NMS Communications

#### 5.6.1.5.8 Send TT

Generates touch-tones. This can be used to automate a call or to communicate with other telephone services.

### PROPERTIES

**Touch tones to play:** The touch tones to generate. Note the following:

- May include: '1 2 3 4 5 6 7 8 9 0 A B C D \* #' e.g.: 123 or @TONES
- Variables should be populated with characters from the above list only.
- Characters not in the list above will be ignored/skipped.

### ACTION STEPS

None

### SUPPORTED TELEPHONY HARDWARE PLATFORMS

Ai-Logix  
Dialogic  
NMS Communications

### RELATED STEPS



[Flash Hook](#)

#### 5.6.1.6 Data/XML Steps

The following list is comprised of all of the database-related and XML handling steps that are available in TeleFlow Designer. These steps are found under the Data/XML tab of the [Steps Toolbox Window](#).

#### Database Steps

[DB Connect](#)  
[DB Disconnect](#)  
[DB Lock](#)  
[SQL Begin Transaction](#)  
[SQL End](#)  
[SQL End Transaction](#)  
[SQL Fetch](#)  
[SQL Statement](#)

#### XML Steps

[XML End](#)  
[XML Fetch](#)  
[XML Load](#)  
[XML Write](#)

## SQL and XML

As a standard that is frequently extended for proprietary purposes, the breadth of SQL (and its many proprietary changes) is well beyond anything that we could hope to cover in this manual. Rather than attempting to provide tutorials and notes on SQL in this manual, we encourage you to seek this from more appropriate sources. There are plenty of courses, books, and online tutorials for everything you might hope to accomplish with SQL and databases. Frequently, you need to learn not just "the basics" of SQL, but also the specifics of how it is implemented in the database product you choose.

XML is a W3C recommendation designed to describe data. Much like SQL, this manual doesn't attempt to recreate the recommendation.

W3 Schools ([www.w3schools.com](http://www.w3schools.com)) has free tutorials and documentation on both SQL and XML, and can be an excellent place to get started with these, and many other useful standards.

### 5.6.1.6.1 DB Connect

Connects to a database using ODBC (This is the only method of connecting to a database from within a TeleFlow Application).

A TeleFlow Application must connect to a database before running any SQL statements or stored procedures (with the [SQL Statement](#) step) to communicate with it. An application can connect to a data source prior to waiting for a call and maintain the connection at all times. TeleFlow does not allow you to have a connection to multiple data sources/databases at one time.

## PROPERTIES

**ODBC Source:** A drop-down list of ODBC data sources currently available/configured on the system. When one is selected, the 'DSN =' line of the connect string is sent to the Connect Parameters box.

**Connect Parameters:** The ODBC connect string to use to connect to the database. The database, ODBC drivers the application is using to connect to it, and its settings can all affect what is required in the connect string. Assuming the appropriate DSN has already been set up (See '[Setting up ODBC Data Sources](#)' for help with this.), the following connect commands - with some adjustments for the appropriate user name or DSN name, of course - will provide access to the listed commonly used database systems:

- **dBase:** DSN=dBASEfile;
- **MySQL:** DSN=MySQLServer;
- **SQL Server:** DSN=SQLServer;UID=MANAGER;PWD=password;

Note that most databases - as shown above - require only that the application connect with "DSN=" and the data source name. Some require that the application itself, and not the datasource, supply user names and passwords for security.

## BUTTONS

**ODBC:** Opens the ODBC Data Source Administrator, where you can set up new data sources or modify those already set up.

## ACTION STEPS



**Failure:** A database connection might fail if the database server is currently unavailable, or if

some information in the connect string is incorrect, such as when a user name supplied isn't valid for the database.

#### RELATED STEPS



[DB Disconnect Step](#)



[SQL Fetch Step](#)



[SQL Statement](#)

##### 5.6.1.6.2 DB Disconnect

Disconnects from the currently connected data source (i.e. The one most recently connected to with the [DB Connect](#) step)

#### PROPERTIES

None

#### ACTION STEPS

None

#### RELATED STEPS



[DB Connect Step](#)



[SQL End Step](#)



[SQL Fetch Step](#)



[SQL Statement](#)

##### 5.6.1.6.3 DB Lock

Locks the most recently fetched record so that no other application is able to update it. An attempted update will result in a failure. The lock is released once a SQL End or a SQL End Transaction is encountered.

**Note:** Please refer to the [SQL Users Guide](#) for more information about SQL programming in TeleFlow.

#### PROPERTIES

**SQL handle:** Each subsequent SQL command to the SQL Statement must be assigned a matching number, or handle, in order relate commands together. This number must range from one to nine. e.g.: 1 or @HANDLE

**ACTION STEPS****RELATED STEPS**

## 5.6.1.6.4 SQL Begin Transaction

In order to perform transaction processing, a starting point must be defined so that data can either be committed or rolled-back once database processing is completed. This starting point is a marker that database activity can be restored to.

**Note:** Please refer to the [SQL Users Guide](#) for more information about SQL programming in TeleFlow.

**PROPERTIES**

None.

**ACTION STEPS**

None.

**RELATED STEPS**



#### 5.6.1.6.5 SQL End

This step ends/closes a SQL handle created by the [SQL Statement](#) step. Although it may be a "good practice" to end all SQL handles created in a TeleFlow Application when it is finished with them, it is not strictly necessary. When a SQL Statement step uses a handle that is already in use, it ends the existing handle automatically first, freeing it for use for the new statement. See the [SQL Statement](#) step topic for more information on SQL handles.

**Note:** Please refer to the [SQL Users Guide](#) for more information about SQL programming in TeleFlow.

#### PROPERTIES

**SQL handle (1..9):** The SQL handle of the statement to end.

#### ACTION STEPS

None.

#### RELATED STEPS



#### 5.6.1.6.6 SQL End Transaction

Ends the transaction defined by SQL Begin Transaction. The database can be returned to the state it was in before the SQL Begin Transaction, or the recent changes can be committed, or saved permanently.

**Note:** Please refer to the [SQL Users Guide](#) for more information about SQL programming in TeleFlow.

#### PROPERTIES

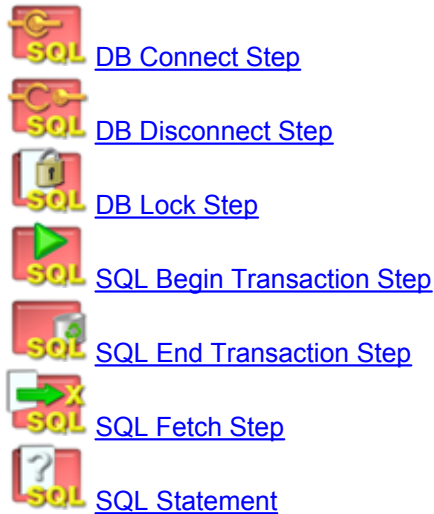
**Parameters:** Determines whether the transaction defined by the SQL Begin Transaction are committed or rolled back, based on which radio button option is selected. The radio buttons are:

- Commit / Save changes to database
- Rollback / Discard changes

### ACTION STEPS

None.

### RELATED STEPS



#### 5.6.1.6.7 SQL Fetch

Retrieves/fetches a single record/row of results of a [SQL Statement](#) step (running a SQL Fetch for a SQL Statement that doesn't generate results - such as an UPDATE - will generate a [Fatal Error](#)). Each time a SQL Fetch is run, it retrieves the next row of results for the associated SQL Statement step, failing if there are no more rows.

**How the results are retrieved:** The row results are stored in TeleFlow variables, with each field or alias defined in the SQL Select statement populating a variable. If a variable with a name matching one of the fields exists, that variable is used to store the result. Otherwise, a global variable of that name is created and populated when the SQL Fetch is run. (Since TeleFlow treats all variables as UPPER CASE regardless of how they are addressed in an application, the newly created variables are created upper case) For the following example SQL Select statement, the TeleFlow Variables @FIRST\_NAME, @LAST\_NAME, and @EMAIL\_ADDRESS would be created and populated or populated when the SQL Fetch for the statement runs, assuming there are any rows matching the criteria:


```
SELECT
  first_name,
  last_name,
  email AS EMAIL_ADDRESS
FROM
  customer
WHERE
  user_id = 12345
```

**Note:** Please refer to the [SQL Users Guide](#) for more information about SQL programming in TeleFlow.





## PROPERTIES

**SQL handle (1..9):** The SQL Handle of the [SQL Statement](#) step to fetch a row for. This is how a SQL Fetch step is related to a SQL Statement step. Typically, a SQL Statement step that has a result set is followed immediately by a SQL Fetch in the application flow. Handles can range from 1 to 9. e.g.: 1 or @HANDLE

## ACTION STEPS

 **Failure:** A SQL Fetch failure occurs when there are no rows to retrieve from the database. This occurs when the SQL Statement the fetch is related to doesn't result in any rows matching the statement's criteria, or when repeated uses of the Fetch step run out of matching rows. This means that a failure on a SQL Fetch step doesn't necessarily mean anything has gone wrong, although it might, under the right circumstances.

## RELATED STEPS

-  [DB Connect Step](#)
-  [DB Disconnect Step](#)
-  [SQL End Step](#)
-  [SQL Statement](#)

### 5.6.1.6.8 SQL Statement

Executes a SQL statement or stored procedure.

**Note:** Please refer to the [SQL Users Guide](#) for more information about SQL programming in TeleFlow.

## PROPERTIES

**SQL handle (1..9):** Every SQL statement must be assigned a number, or handle, which other database steps use to refer to or work with the statement or its results (E.g. To fetch a results row from a SQL Statement, you would use a [SQL Fetch](#) step, and supply the handle from the step that generated the results set.). This number must range from 1 to 9.

**SQL statement:** The definition of the SQL statement. TeleFlow runs SQL statements through an ODBC driver, so the driver determines what you can and cannot do (i.e. The driver determines what SQL statement syntax and database functions work). The following example shows what a select statement might look like. TeleFlow variables are populated in the 'select' section, and used to locate information in the where clause. The following select statements was created for dBase tables.

```
SELECT
  acct1 + acct2 + acct3 ACCTOT,
  shipto SHIPTO,
  shipdate SHIPDATE
FROM
```



```
\DATA\Account.dbf
WHERE
acctno = @ACCTNO
```

**Time warning message threshold (ms):** Whenever TeleFlow runs a SQL Statement step, it notes how long it takes - in milliseconds - for ODBC to execute it and return a result. A warning message is displayed in the log output if this threshold is crossed. The elapsed time, in seconds, is output to the TeleFlow log, following the output from the SQL Statement step itself, like so:

```
Elapsed Time: 0.015
```


If the time is greater than the threshold amount supplied, a warning message will follow the elapsed time, as shown:

```
Elapsed Time: 3.220 - TIME WARNING
```

As this warning message is consistent, it can be searched for in TeleFlow log files to easily locate SQL statements that are in need of tuning.

If no value is supplied, an internal default of 2000 (2 seconds) is used. Note that the value in this field does not change anything about the execution of the SQL statement.

## ACTION STEPS

 **Failure:** The primary reason to take advantage of the fail path on a SQL Statement step is to handle issues with database connectivity. Sometimes, for one reason or another, a database you connected to with the [DB Connect](#) step is unavailable when you attempt to run a SQL Statement on it. It is sometimes possible - depending on how long the database remains inaccessible - to recover from such a problem if you put a short pause - using the [Wait](#) step - after a failure, then try the SQL Statement again. Because the failure could also be incorrect syntax, and because a database might remain unavailable, it is a good idea to retry only a few times before giving up and generating a fatal error (using the [Fatal Stop](#) step). Many of the sample applications that come with TeleFlow Designer use flowcharts named "DBSet" and "DBRetry", which have examples of how to go about building such retries into a TeleFlow Application.

## RELATED STEPS



### 5.6.1.6.9 XML End

Closes an XML handle and deletes the associated XML document from memory.

An XML document is referenced by a specific number, or handle, as set by the [XML Load](#) step that loaded the document. Once you are finished with an XML document, you should close its handle in order to free that memory.

## PROPERTIES

**XML handle:** The number of the XML handle to close. This number must be one previously assigned in an XML Load Step.

## ACTION STEPS

None.

## RELATED STEPS:



### 5.6.1.6.10 XML Fetch

Retrieve a value and/or attribute values from a node of an XML document.

To use this step, you must have loaded an XML document into memory with the [XML Load](#) step.

## PROPERTIES

**XML handle:** The number of the XML handle to fetch matching nodes from. This number must have been assigned by an [XML Load](#) step.

**Node:** The name of the xml node to retrieve. As XML nodes are hierarchical in nature, the node must be specified in notation that indicates where it is in relation to other nodes in the document. The TeleFlow notation for this follows:

`{document node}->{node}->{node}...`

or:

`{node}->{node}...`

where:

*{document node}* is the XML document node. As all XML documents have a document node, this aspect of referencing the node is optional.

*{node}* is any XML node that exists in the document. Note that if a node is specified as a "sub-node", it will only be located if it is a sub-node of the node (or nodes) indicated. When running multiple fetches, it is important to know that each node search is run from a position just after the most recently fetched node. Note that a node could be a parent node to the next node, and/or a child node to the previous node. You can specify any number of nodes.

The following are examples of valid XML node specifications

```
xml doc->employees->emp
employees->emp
employees
PRODUCE->Fruit->Apples
Fruit->Apples
Fruit
```

**Capture value to variable:** If the node of an XML document has a value, capture it to the TeleFlow® variable supplied in this field. This is an optional property. If this field is left empty, the value of the node will not be captured. Note that if a variable is specified here, but the node does not have a value the XML Fetch will fail (and take the path specified by the Fail action step).

**Attributes:** The node attributes to retrieve. Place each node name to retrieve on a single line. TeleFlow® will assign the attribute's value to a TeleFlow® variable with the same name. Optionally, specify the variable to assign the value to with the "as" keyword, for example: "Name as EMP\_NAME". Do not include the '@' symbol in the variable name.

**Fetch direction:** Fetch forward for the next occurrence of a node, or backward for the previous occurrence. Each time an XML Fetch is run for a particular XML handle, it continues from where the last fetch left off. This means that if there are 4 matching nodes, and you start fetching forward, you can run a forward fetch for the handle 4 times before the application flow will follow the Failure/bomb path. This also allows you to go backward and forward through nodes in an XML document. This can be helpful when playing fetched results to a caller one at a time: If the caller is able to go forward and backward to the next and previous results, you can have two different XML Fetches using the same XML Handle, one set with forward fetch direction, the other backward, allowing the caller to "navigate" through the data.

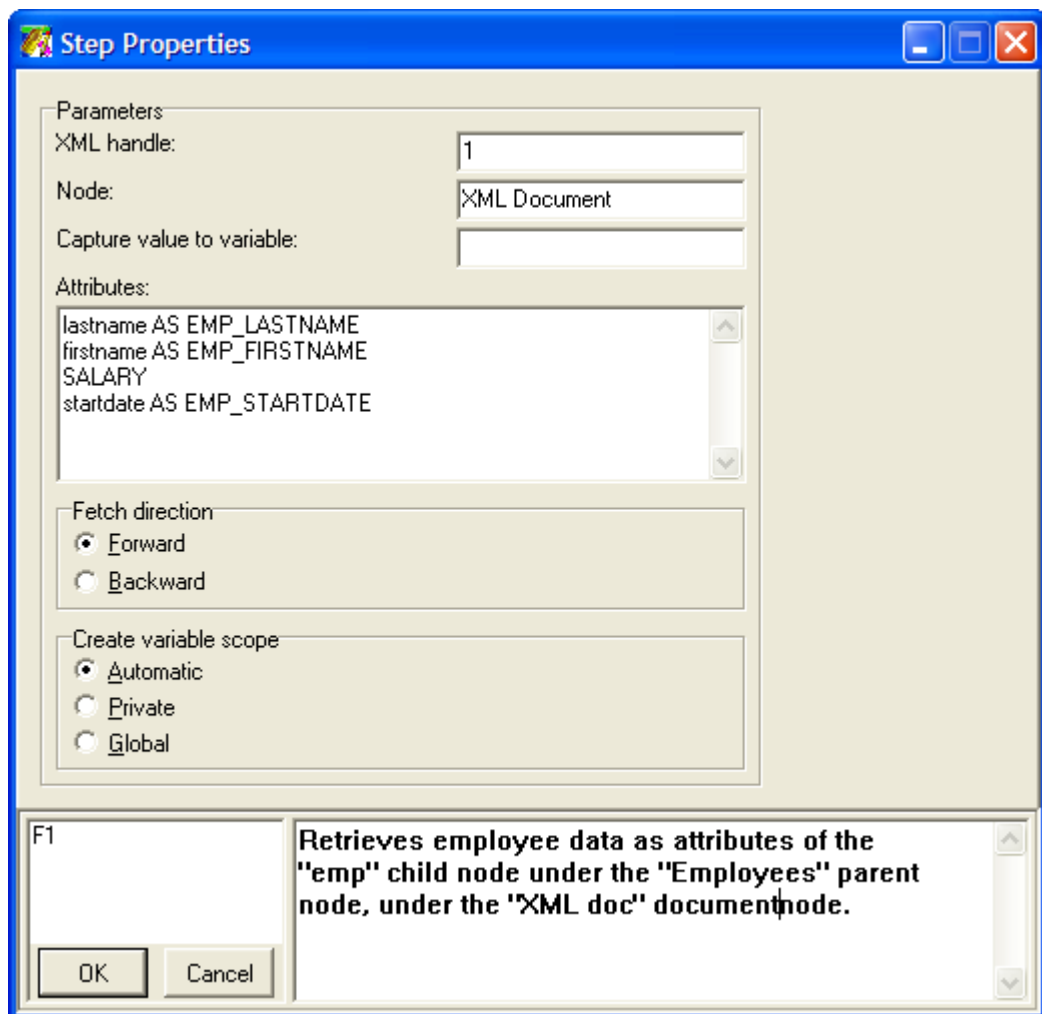
**Create variable scope:** Indicates how TeleFlow® assigns (or creates) variables to store the node value and attribute values. The default behavior, "Automatic", uses local variables first if they already exist, then uses global variables, and creates any new variables as global variables. "Local" only uses and creates local variables, and "Global" uses and creates only global variables.

## ACTION STEPS



**Failure:** TeleFlow® could not find the XML node requested. This could occur because the node does not exist, or because there are no more instances of the node available to fetch (in the forward direction), or attempted to fetch before the first node (in the backward direction). When a failure occurs, the "search position" in the XML document for the XML handle is either at the beginning (if the Fetch direction was backward), or the end (if the Fetch direction was forward).

## EXAMPLE

**RELATED STEPS:**

-  [Load Webpage](#)
-  [XML Load](#)
-  [XML Write](#)
-  [XML End](#)

## 5.6.1.6.11 XML Load

Load an XML document into memory.

Before an XML document can be manipulated by TeleFlow®, it must be loaded into memory. An XML document can come from many sources. The most common are from a file on disk or from a separate XML server (likely a web server, using an HTTP Get). Accordingly, the XML Load step supports two sources: a file, or TeleFlow® variable. A TeleFlow variable might get populated with XML in a number

of different ways, although the most common would be the HTTP Response Message Body from an [HTTP](#) step.


## PROPERTIES

**XML handle:** Every XML document loaded into memory must be assigned a number, or handle, in order for all subsequent XML Steps to reference it (The [XML Fetch](#) step uses this handle to retrieve nodes and attributes from the XML document). This allows more than one XML document to be loaded into memory at a time. The XML handle must be numeric, but can be contained in a TeleFlow® variable (eg: @HANDLE).

**Load From:** Specifies the source type of the XML document. If this option is set to "Variable", you must supply a TeleFlow® variable in the "Variable / File to load" property. If set to "File", you must supply a file name in the "Variable / File to load" property.

**Variable / File to load:** Indicates the actual TeleFlow® variable or file name - the filename can be in a TeleFlow variable, and the path to the filename is also required - containing the XML document to be loaded. The value required for this field varies depending upon the setting of the "Load From" property (see above).

## ACTION STEPS

 [Failure](#): TeleFlow® could not find the XML document requested, or the document doesn't appear to conform to proper XML document format.

## RELATED STEPS:



### 5.6.1.6.12 XML Write

Writes an XML document currently stored in memory to a disk file.

## PROPERTIES

**XML handle:** The number of the XML handle of the XML document to be written to disk. This must be a number already assigned by an [XML Load](#) step.

**File Output:** The file to write the XML document to. The file name specified should contain the path to where the file should be created (eg: C:\My Documents\Sample.XML). Otherwise, the file will be created in the current working directory that TeleFlow® Server is running from.

**Append:** If checked, the XML document will append to an existing document. By default this is

unchecked, indicating that an existing file should be over-written by the new XML document.

### ACTION STEPS

None.

### RELATED STEPS:



#### 5.6.1.7 Documentation Steps (Doc. tab)

The following list is comprised of all of the Document steps that are available in TeleFlow Designer. These steps are found under the Doc. tab of the [Steps Toolbox Window](#).

There are different "types" of documentation steps in TeleFlow. Some affect flowchart readability, others document the application, and some provide the developer the ability to add information to or make special copies of the TeleFlow log.

[Connect Point](#)  
[Notes](#)  
[Save Log](#)  
[Step Comments](#)  
[Write Text](#)

##### 5.6.1.7.1 Connect Point

This step allows a flow line to be connected to the next step via a more specific route than just a direct connection. It is extremely handy as lines can be drawn more tidily away from and around other steps. This helps to keep connecting lines from crossing over other steps, making flowcharts easier to read (which is why it is referred to as a documentation step: In a flowchart-based development environment, a clean, easy-to-follow flowchart is as much a part of documentation as comments on why something is being done are.)

**NOTE:** Unlike other TeleFlow steps, you can drop new steps on top of Connect Point steps, replacing them with the newly dropped step. The flow shape leading to and from the connect point being replaced is kept for the replacement step.

### PROPERTIES

None.

### ACTION STEPS

None.

#### 5.6.1.7.2 Notes

TeleFlow Application flowchart notes. Notes typed into this step are visible in the TeleFlow flowchart space, and the box around them is sizeable, allowing the application developer to surround a group of steps with a comment that applies to all them.

This step is not connected to any of the flowchart steps, and is purely for documentation purposes.

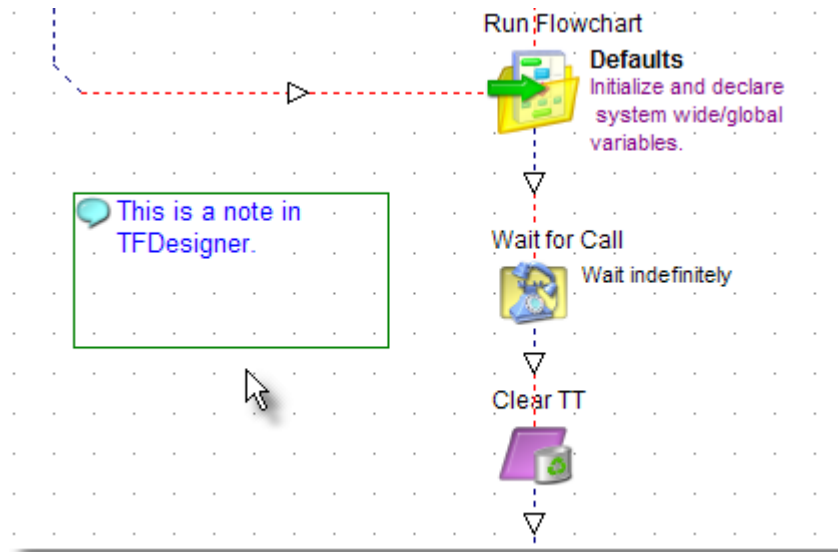
#### PROPERTIES

None.

#### ACTION STEPS

None.

#### EXAMPLE



#### 5.6.1.7.3 Save Log

Saves the contents of the TeleFlow application log to a file on disk.

#### PROPERTIES

**Save to TeleFlow Log folder:** This option uses the current TeleFlow Log directory setting as the location to save the log output. See [TeleFlow Config](#) for more information on setting the TeleFlow log directory.

**Save to folder:** This options saves the log to the directory specified as long as it is a valid Windows path. The value provided has no bearing on the current TeleFlow log directory setting in [TeleFlow Config](#). The setting takes affect for this step only.

**Use TeleFlow Log name:** This option uses the name associated with TeleFlow Application (TAP) as the file name to write the log information to. The name is defined when creating a list of applications to run in [TeleFlow LineList](#).

**Specify file name:** This option specifies the name of the file to log to. Provide a valid Windows filename.

## ACTION STEPS

None.

## RELATED STEPS



[Write Text](#)

### 5.6.1.7.4 Step Comments

Every step in TeleFlow Designer has both automatic and developer-defined comments.

**Automatic Comments:** As step properties for steps in an application are defined, key data from these properties are displayed in the flowchart space. Each step has the step name just above it, and the key data from its properties off to the right of the step. Both of these data are in black text. In this way, a TeleFlow Application is said to be "self-documenting", because much of what you would want to know about a given step is shown right in the application/call flow.

**Developer-defined Comments:** The step properties window for every step has a field for developer comments on the bottom right-hand side. These comments appear in the flowchart space in blue text, beneath the automatic comments. Use these comment sections to make specific notes about what the step is for, unless it is obvious enough without it. (E.g. A Play step that plays a script code that has a complete script showing tells anyone reading the application exactly what is going on.)

### 5.6.1.7.5 Write Text

Outputs text to the TeleFlow log or a text file.

## PROPERTIES

**Message text:** The text to output. This can be a combination of standard text and variables. e.g.:  
Current account: @AcctNo Balance: @AcctBal

**Start new line after message text:** If checked, a carriage return is included at the end of the text. This means that if/when another line of text is output to the same file, it will start on a new line. In order to construct one message line with several message commands, do not check the start new line box.

**File output:** If defined, the message text output will be directed to a file. Otherwise, the Message text is output to the TeleFlow log. e.g.:c:\ivrapp\txt\textfile.txt or @TextFile

**Append:** If checked, the message text will append to the file if it exists, or create a new file if it doesn't. If unchecked, the file will be overwritten if it exists, and created if it does not.



## ACTION STEPS

None.

## RELATED STEPS



[Load Text](#)

### 5.6.1.8 Programming Steps (Prog. tab)

The following list is comprised of all of the programming steps that are available in TeleFlow Designer. Programming steps in TeleFlow cover logical operations, file manipulation, variable handling, global events, labels, scripting, logging, running flowcharts/programs/apps, and calculations.

These steps are found under the Prog. tab of the [Steps Toolbox Window](#).

- [Case](#)
- [Compare](#)
- [Copy File](#)
- [Create Folder](#)
- [Date Info](#)
- [Delete File](#)
- [DLL Function](#)
- [Encrypt](#)
- [Fatal Stop](#)
- [Financial Calc](#)
- [Global Event](#)
- [Go To Label](#)
- [Hash](#)
- [Label](#)
- [Load Text](#)
- [Math](#)
- [Parse](#)
- [Random Number](#)
- [Read Registry](#)
- [Reload Off](#)
- [Reload On](#)
- [Run Basic Script](#)
- [Run Flowchart](#)
- [Run Program](#)
- [Run TAP](#)
- [Save Log](#)
- [Scan Folder](#)
- [Set Global Variable](#)
- [Set Language](#)
- [Set Local Variable](#)
- [Set Play Variable](#)
- [Wait](#)
- [Write Text](#)

#### 5.6.1.8.1 Case



Compares one value to up to eight others, seeking the first matching option, and following the path set for that option, or that set for "no match" if none is found.


## PROPERTIES

**Is argument:** The value to compare against the values in fields 1 through 8. e.g.: @GETYN or 2001/01/22

**1-8:** Values to be compared against the "Is argument" entry. When a match between the two is found, the path set for the matching numbered entry is followed.

## ACTION STEPS

 **through** : If the "Is argument" value matches with one of the numbered values, the corresponding numbered path is followed.



: If no matches between "Is argument" and the numbered values is found, the ?, or "no match" path is followed.

## RELATED STEPS



[Compare](#)

### 5.6.1.8.2 Compare

Compares one value to another, determining if the comparison expression defined is true or false. If the expression in the Compare step is true, the path set by the  True action step is followed. Otherwise, the path set by the  False action step is followed.

## PROPERTIES

**Is value:** The value to compare against the second value. e.g.: @GETYN or 2001/01/22

**Function:** Defines the way in which to compare one value with another. The possible comparison functions/operators are:

- = (Compare to see if "Is value" is equal to "Value")
- <> (Compare to see if "Is value" is NOT equal to "Value")
- > (Compare to see if "Is value" is greater than "Value")
- < (Compare to see if "Is value" is less than "Value")
- >= (Compare to see if "Is value" is greater than or equal to "Value")
- <= (Compare to see if "Is value" is less than or equal to "Value")
- Empty (Is the "Is value" entry empty/contain nothing?)
- Modulus 10 (the modulus 10 comparison performs a check to determine if the "Is value" number is a valid modulus 10 (LUHN) number (i.e. a valid credit card number, a valid Canadian social insurance number, etc.))
- Valid (i.e.: date, time, number) The Valid option is used to ensure that "Is value" conforms to a variable format. By using this function, data can be validated before being processed. This is useful to ensure that date or time information entered be a caller is valid.

**Value:** The value to compare against the first value. e.g.: @TODAY or Y

## ACTION STEPS



True: If the comparison evaluates to 'True', this path is followed.



False: If the comparison evaluates to 'False', this path is followed.

## RELATED STEPS



[Parse](#)

### 5.6.1.8.3 Copy File

Copies a specified file from one directory path and filename to another. In a voice mail application, this step could be used to move a recorded voice message to another directory once it has been listened to.

**NOTE:** This steps does not support wildcard character use. (I.e. You cannot copy \*.Wav from one directory to another)

## PROPERTIES

**Copy filename:** Directory path to and filename of a file to copy. e.g.: c:\ivrapp\vmmailincoming.vox or @VOXFILEFROM

**To:** Directory path and filename to copy the file to. e.g.: c:\ivrapp\vmmail\heard.vox or @VOXFILETO

## BUTTONS

**Locate:** Opens a file location dialog window. Use this button to find a file.

## ACTION STEPS



**Fail:** The copy failed. This may occur for a number of reasons, but the most likely causes are that the file was not located, or it was opened exclusively by another process. Invalid file names will also cause the process to follow this path.

## RELATED STEPS



[Delete File](#)



[Record](#)

#### 5.6.1.8.4 Create Folder

Creates a directory/folder on disk. A directory can be created on any drive and within any other directory.

#### PROPERTIES

**Create directory (folder):** A valid directory path to create. Create Folder will not create multiple directories in a path. (I.e. All but the lowest sub-directory in the path must already exist.) e.g.: C:\VoiceMail\NewDirectory or @NewDirectory

#### ACTION STEPS



**Fail:** The directory creation failed. This may occur for a number of reasons, but the most likely cause is that the directory path/name was not valid.

#### RELATED STEPS



[Scan Folder](#)

#### 5.6.1.8.5 Date Info

Creates or sets a variable to a textual description of an aspect of the date, such as the day of the week. The step offers several options as to the final format of the date name value. The variable can be set to provide the name in full or in the standard 3-letter abbreviation, in addition to regular or all upper case. For example: The Short day name of date '2001/01/22' in regular case is 'Mon'. The Long month name of '2001/01/22' in all upper case is 'JANUARY'. See the [Variable Usage](#) section for more information on variables.

#### PROPERTIES

**Variable to receive result:** The name of the variable to set or define with the date name. e.g.: @DATENAME

**Function:** The modifying effect on the 'of date' value. Possible modifying effects:

- Short day name (eg: Mon)
- Long day name (eg: Monday)
- Short month name (eg: Jan)
- Long month name (eg: January)

**of date:** Variable or literal containing the TeleFlow date value to modify and set/define the 'Result' to/as. (For example: '@DATE')

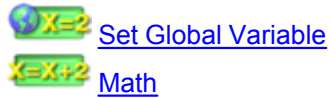
**Return as:** The formatting to apply to the date-name. Possible modifying effects:

- Regular case (eg: Monday)
- UPPER CASE (eg: MONDAY)

#### ACTION STEPS

None.

## RELATED STEPS



### 5.6.1.8.6 Decrypt

Returns encrypted information to its original form. The key value that the text was originally encrypted with is required for decryption. If TeleFlow® was used to encrypt the data, the key will be the same value supplied with the Encrypt Step's "Encryption key" property.

## PROPERTIES

**Encrypted text:** The encrypted version of the text to decrypt. As encrypted text is not easily typed, it is likely that you will have the encrypted text stored in a single TeleFlow® variable.

**Decryption key:** Key to use to decrypt the text. This key must match the key that was originally used to encrypt the data.

**Variable to accept decrypted text to:** Specify a variable to receive the decrypted version of the text. Once the step has completed successfully, this variable will contain the entire text in its original form.

## ACTION STEPS

None.

## RELATED STEPS:



## NOTES:

If you use another program *besides* TeleFlow to encrypt your data, it is important to know the following.

The encryption mechanism used by TeleFlow is Blowfish in Cipher Block Chaining (CBC) mode.

Because TeleFlow variables hold strings only, it does not have a storage type for binary data. You need to provide TeleFlow with encrypted data in hexadecimal format. This data must be in the form of one long string with each two-character pairs representing a single byte of binary data. The first two characters are the hexadecimal value of the first byte of encrypted data; the third and fourth characters are the hexadecimal value of the second byte of encrypted data; and so on.

For more information on the Blowfish algorithm can be found at <http://www.schneier.com/blowfish.html>

#### 5.6.1.8.7 Delete File

Deletes a file from location specified. This is commonly used in voice mail applications to delete files when a caller opts to re-record a message.

**NOTE:** This steps does not support wildcard characters. (ie. You cannot delete \*.\* from a directory)

#### PROPERTIES

**Filename:** Directory path and name of the file to delete. e.g.: c:\ivrapp\vmail\myfile.vox or @VMAIL\_FILE

#### BUTTONS

**Locate:** Opens a file location dialog window. Use this button to find the file to delete.

#### ACTION STEPS



**Fail:** The delete failed. This may occur for a number of reasons, but the most likely causes are that the file was not located, or it was opened exclusively by another process. Invalid file names or directory paths will also cause the process to follow this path.

#### RELATED STEPS



[Copy File](#)



[Record](#)

#### 5.6.1.8.8 DLL Function

Calls a function from a custom dynamically linked library (DLL).

Use of this step requires a custom DLL that exports a function with the following properties (see example):

- accepts a constant char\* null terminated string as the sole parameter
- returns type char\* that points to a dynamically allocated null terminated string that can be freed by the step

#### PROPERTIES:

**DLL file containing function:** The directory path to and filename of the DLL file from which the function will be called.

**Function name to call:** The name of the function, with underscores if necessary.

**Function parameters:** The string of information that will be passed to the function.

**Variable to receive result:** The TeleFlow variable that will receive the string returned by the function; this variable will be empty if the function returns a NULL pointer

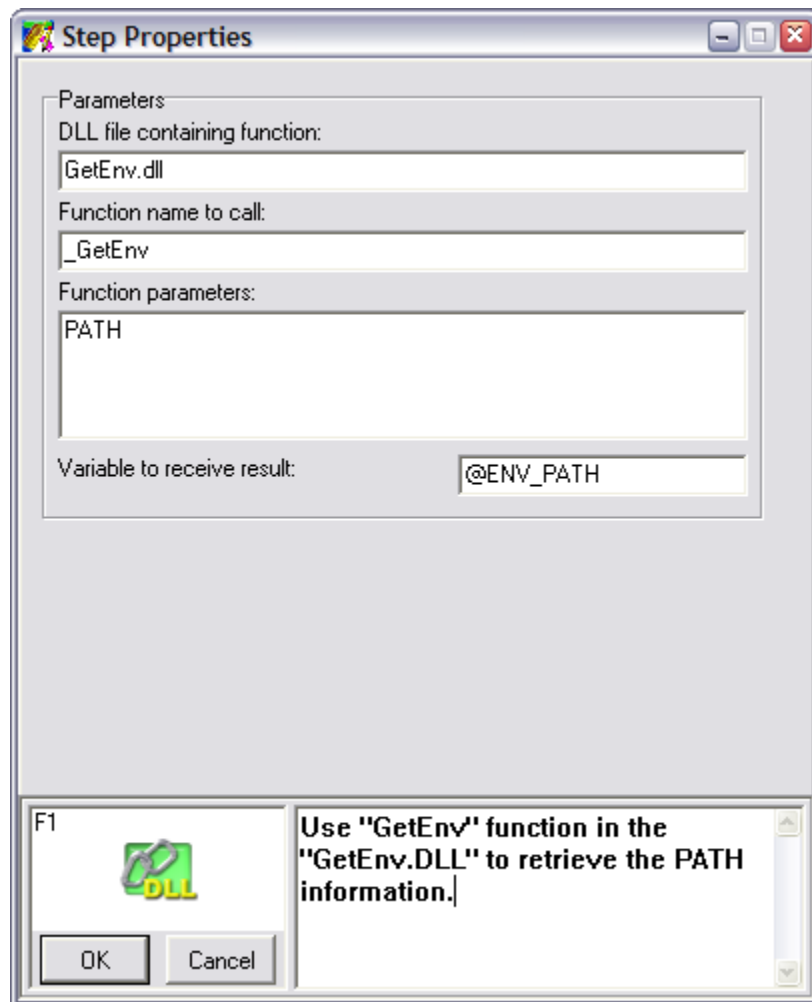
#### ACTION STEPS:



**Failure:** Here are some reasons the DLL Function step might fail:

- TeleFlow could not find the DLL file: the path may not have been correct; the file extension may have been omitted.
- A function matching the name with the correct return type and parameters was not available from the DLL: an underscore may have been necessary; the function may not be exported properly.
- Sometimes this step will fail if too many application instances run it on the same DLL at one time. For this reason, it can help on a Fail to wait for a second, then try again (although, this should only be done once or twice for an individual step, to avoid infinite loops).

**EXAMPLE:** The following screen capture is from the properties step of the DLL Function step in the example "DLL Function" application that ships with TeleFlow.



Get Environment Variable: getenv.cpp and getenv.h.

**RELATED STEPS:**

None.

## 5.6.1.8.9 Encrypt

Encrypts text so that cannot be decrypted without the appropriate decryption key. This feature is typically used to transfer information across an insecure medium, such as the Internet.

**PROPERTIES**

**Text to encrypt:** The text to be encrypt into an unreadable string. The text can contain TeleFlow® variables, which will be replaced before the text is encrypted.

**Encryption key:** The key to use to encrypt the text. The key can be any string of characters, and is required to decrypt the text.

**Variable to accept encrypted text into:** Specify a variable to receive the encrypted text into. Once the Step has completed successfully, this variable will contain the entire text in encrypted form.

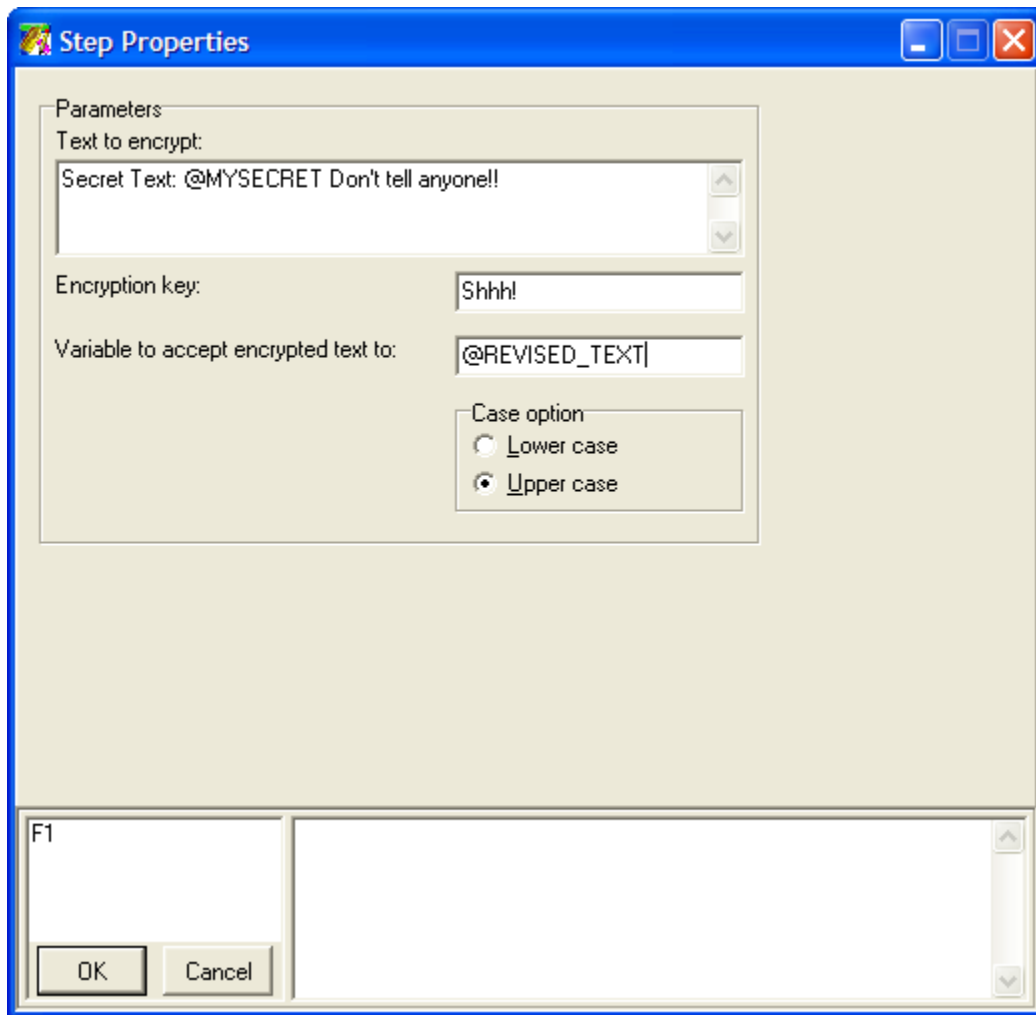
**Case option:** Indicates the case (lower or UPPER case) of the output hexadecimal sequence.

**ACTION STEPS**

None.

**EXAMPLE:**





The image shows a 'Step Properties' dialog box with a blue title bar. It contains the following fields and options:

- Parameters:**
  - Text to encrypt:** A text area containing 'Secret Text: @MYSECRET Don't tell anyone!'.
  - Encryption key:** A text box containing 'Shhh!'.
  - Variable to accept encrypted text to:** A text box containing '@REVISED\_TEXT'.
  - Case option:** A group box containing two radio buttons: 'Lower case' (unselected) and 'Upper case' (selected).
- Footer:** A small text box containing 'F1', and 'OK' and 'Cancel' buttons.

**RELATED STEPS:****NOTES:**

If you use another program *besides* TeleFlow to decrypt your data, it is important to know the following:

The encryption mechanism used by TeleFlow is Blowfish in Cipher Block Chaining (CBC) mode.

Once TeleFlow encrypts your text, the encrypted data is binary, and TeleFlow does not have a storage type for binary data. TeleFlow variables hold strings only. To work around this issue, TeleFlow stores the binary data in a string of 2-character hexadecimal bytes. The first two characters are the hexadecimal value of the first byte of encrypted data; the third and fourth characters are the hexadecimal value of the second byte of encrypted data; and so on.

In order to decrypt your data, you must first convert it from hexadecimal back to binary. Then you can decrypt it to the original string using any implementation of the Blowfish algorithm in CBC mode, as long as you have the key you used for the original encryption.

For more information on the Blowfish algorithm can be found at <http://www.schneier.com/blowfish.html>

#### 5.6.1.8.10 Fatal Stop

Forces the TeleFlow Server/Simulator to stop the application immediately, generating a [Fatal Error](#) event.

#### PROPERTIES

None.

#### ACTION STEPS

None.

#### RELATED STEPS



[Label](#)



[Global Event](#)

#### 5.6.1.8.11 Financial Calc

Provides mortgage estimates based on income, interest, and mortgage amounts.

Note: It should always be noted to callers and users of the system that the amounts provided are only approximations, and that only a lending institution can provide accurate final numbers. There are many factors that go into determining these amounts, and there are many options where mortgages and payments are concerned.

#### PROPERTIES

**Variable to receive result:** The variable to receive the monthly payment or mortgage amount.

**Determine:** The type of mortgage calculation to make, with the following options:

- **Monthly payment:** The approximate monthly mortgage payment one would have to make based on the "Annual interest rate" and "Mortgage amount" entries.
- **Mortgage amount:** The maximum mortgage amount one could expect to be given by a lending institution based on the total amount of the mortgage to "Annual interest rate" and "Gross income" entries.

**Annual interest rate:** The expected annual interest rate for the mortgage to be acquired.


**Mortgage amount:** The total amount of the mortgage.

**Gross income:** A person's gross annual income.

## ACTION STEPS

None.

### 5.6.1.8.12 Global Event

Sets the [Labels](#) to go to for special events that may occur during the operation of an application. At any time during the flow of the application, events can be directed to different  [Labels](#). When one of the set events occurs, the TeleFlow Server or Simulator will search the current flowchart, then the previous flowchart (i.e. the flowchart that ran the current flowchart), and so on, essentially backtracking the direction the user came, for the first instance of the Label name defined. This is where the Server will resume the flow of the application (i.e. When the Label step is found, the application will continue from it). An important function of the Global Event step is to indicate where to continue running from when the caller hangs up. Often, logging will be placed at the Global Hang-Up label to track call duration and

other aspects of the call before returning to a  [Wait For Call](#) step to wait for the next call to come in.

When defining a global label, it is best to ensure that the matching Label step is in the same flowchart as it is being defined in. (If for no other reason than it is easier to read your application this way) It is also acceptable to set the global label to a label that is certain to exist in a flowchart higher up in the application/call flow.

See the [Go to Label](#) topic/step for more information about how labels work in TeleFlow.

## PROPERTIES

**The type of event:** Sets the type of event to define a label for, per the following options:

- **Hang-up:** When the user hangs up, go to the label name defined.
- **Fatal error:** When an error occurs that the TeleFlow Server/Simulator is not able to handle, or the application reaches a [Fatal Stop](#) step, go to the GLOBAL label name defined.

**GLOBAL label to go to for event:** The name of the label to jump to when the event set above occurs. e.g.: START

## ACTION STEPS

None.

## RELATED STEPS



## 5.6.1.8.13 Go To Label

Instructs the application to jump to and continue its flow from the defined  [Label](#) step.

TeleFlow searches for a Label in the current flowchart first, then the previous flowchart (I.e. the flowchart that ran the current flowchart), and so on, essentially backtracking the direction the application came to reach the point at which the Go To Label step was run, for the first instance of the Label name defined. This is where the Server will resume the flow of the application (I.e. When the Label step is found, the application will continue from it).

In a manner of speaking, you could say that a go to label and the label to which it leads equate to a flow line in the flow chart. You can't see the line physically, but it connects the two steps nonetheless. This line exists between the steps even if you have to jump back through flowcharts to find it. If TeleFlow finds its way back to the first flowchart run without finding the label, it generates a FATAL error.

**PROPERTIES**

**Go to label:** The name of the label step to go to. e.g.: MAIN\_MENU

**ACTION STEPS**

None.

**RELATED STEPS**

## 5.6.1.8.14 Hash

Generates an MD5 hash of a given string.

The hashing method used is the "RSA Data Security, Inc. MD5 Message-Digest Algorithm", and is provided under license. If you are using TeleFlow to create an application for resale, you are required to abide by the same terms. See "NOTES" below. The MD5 hash value is not reversible. You can create an MD5 hash, and use that hash to compare against another MD5 hash to determine if the two original values are the same, but you cannot use the MD5 hash to determine what the original value was. For this reason, the most common use for MD5 hashing is to store sensitive information, such as passwords.

**PROPERTIES**

**Text to hash:** The text or variable containing text to be hashed.

**Variable to accept hashed text to:** Specify a variable to receive the hash into.

**Case option:** Indicates the case of the output hexadecimal sequence (lower case or UPPER CASE).

**ACTION STEPS**

None.

**RELATED STEPS:****NOTES**

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

## 5.6.1.8.15 Label

Defines a point, or label, anywhere in a [flowchart](#) where a [Go To Label Step](#) can jump to and continue from. Use labels to retain a readable flowchart once a flowchart becomes complex.

The label is also used to define where to jump to when a [Global Event](#) occurs.

**PROPERTIES**

None.

**ACTION STEPS**

None.

**RELATED STEPS**

## 5.6.1.8.16 Load File Step

Loads the contents of a file into a TeleFlow variable, optionally converting to Base 64 encoding to support binary files.

**PROPERTIES**

**Variable to load text into:** The variable that will be created or modified with the contents of the Loaded Text. eg.: @CONTENTS

**From file:** The directory path to and the filename of the text file being opened.

eg.:c:\ivrapp\txt\somefile.txt or @MYFILE

**Variable to receive byte count:** Provide a variable name in this field to capture the total number of bytes loaded. Note that if Base64 encoding is used, this will be the size of the encoded data, not the original size of the binary data on disk. This field is optional, and can be left empty.

**Convert to Base64 (MIME) encoding:** If checked, data in the file will be converted to Base64 encoding when it is loaded. This is typically done if you need to read binary data, however Base64 encoding will also work on text data. If unchecked, the data will not be encoded, and only text will be loaded into your variable.

**Include CR/LF in Base64 count:** In Base64 encoding, CR/LF characters are considered white space, and therefore are not counted by TeleFlow when it stores the converted byte count into the "Variable to receive byte count". If you wish to include the CR/LF characters in the byte count, check this box. Otherwise, leave it un-checked.

## ACTION STEPS



**Fail:** The Load Text failed. The Load Text failed. This may occur for a number of reasons, but the most likely causes are that filename was not valid, or the file does not exist.

## RELATED STEPS



[Write File](#)



[Parse](#)

## NOTES

Encoding and decoding data using Base64 gives TeleFlow the capability to work with binary data. To write binary data use the [Write File](#) step.

### 5.6.1.8.17 Write File Step

Write the contents of a TeleFlow Variable to a file, including support for binary data write.

## PROPERTIES


**Write to file:** Specifies the file to write to. If the file does not exist, it will be created. If this file already exists, it will be overwritten.

The "Locate..." button can be used to find an existing file to be populated into this field.

**Variable containing data:** This field must contain a single TeleFlow variable name that contains all of the data you want to write to the file.

**Decode from Base64 (MIME):** If the variable name specified above contains Base64 encoded data, use this check-box to indicate you want the data to be converted back to binary and written to the file as such. If you leave this box un-checked, the data in the variable will be written as text.

## ACTION STEPS

 **Fail:** A failure can occur if the file specified cannot be written to (such as referencing a folder that does not exist, or attempting to overwrite a write-protected file).

## RELATED STEPS



Load File

## NOTES

Encoding and decoding data using Base64 gives TeleFlow the capability to work with binary data. To get binary data into a variable use the [Load File](#) step.

### 5.6.1.8.18 Math

Performs numeric calculations or string manipulations.

**Note:** Many of the operations that can be done with the Math step can also be done with the [Run Basic Script](#) step. The Run Basic Script has the advantage of being able to perform multiple operations in one step, in addition to being able to run scripts containing loops and conditional code.

## PROPERTIES

**Variable to receive result:** The variable that will be created or modified by the result of the calculations or manipulations. e.g.: @TOTAL

**Base value:** The first variable or value to use in conjunction with the second variable or value to determine the result of the calculation. e.g.: 49 or @APPLES

**Function:** The operation to perform on the Base value and (optionally, depending on the function) Modifying value. Possible operations to perform on these values:

- + (Addition: Base value + Modifying value)
- - (Subtract: Base value - Modifying value)
- \* (Multiply: Base value x Modifying value)
- / (Divide: Base value / Modifying value )
- Modulus of... (Base value Mod Modifying value)
- String append... (Append Modifying value to Base value)
- String length of... (String length of Base value)
- Copy string from... (Copy Base value starting at character number Modifying value for Number of characters to copy)

**Modifying value:** The second variable or value to use in conjunction with the first variable or value to determine the result. e.g.: 51 or @ORANGES

**Number of characters to copy:** If the function is marked as a "Copy string from" operation, then a value must be entered in the field. This entry determines how many characters will be cut to the right of the Modifying value position in the Base value string. e.g.: 5 or @CUTLENGTH

## EXAMPLES

Here are some full examples of entries that the Math step can be used for:

- @TOTAL = 49 + 51
- @TOTAL = @APPLES - @ORANGES
- @NEWSTRING = 'GOOD INFORMATION' Copy string from... 6 ...for... 3 (Result = 'INF')
- @AREA = @WIDTH \* @LENGTH

## ACTION STEPS

None.

## RELATED STEPS



[Run Basic Script](#)

[Set Global Variable Step](#)

[Set Local Variable](#)

### 5.6.1.8.19 Parse

Searches through variables or text looking for the next occurrence of a character, a series of characters, or for a number. The result can be either the character position of the first character of the located occurrence, or the string found in a delimited list.

## PROPERTIES

**Result (not labeled in step properties. It is the first field):** The variable that will be created or modified by a value returned by the parse. If a series of characters or a number is found, the character position they were found in is entered into the result variable. An entry of 0 (zero) will result if the search failed. When using the "Delimiting string..." option, the result will be a string found between delimiters. e.g.: @FINDLOC.

**Parse:** The variable or text to search. e.g.: @LINE or 'The quick brown fox jumps over the lazy dog.'

**Search for:** The type of and data to search the Parse string for, as per the following choices:

- Text string: If selected, a text or variable entry must be entered in the adjoining field. The step searches for text in the "Parse" entry.
- Any character of...: If selected, a text or variable entry must be entered in the adjoining field. The entry must be a space separated list of individual characters to search for, including \* , @ # etc. The step searches for the next encountered single character in the list in the "Parse" entry.
- Delimiting string...: If selected, a text or variable entry must be entered in the adjoining field. The entry must be a character that is used as a "field delimiter" in the "Parse" entry.
- Next real number: If selected, the step searches the "Parse" entry, returning the character position of the next real number, if any.
- Text string if selected, a text or variable entry must be entered in the adjoining field



- Any character of... one or a list of any character to search for, including \* , @ # etc
- Delimiting string... to retrieve the text between the delimiters, such as commas, other characters or even text
- Next real number

**Search from:** Sets where in the Parse entry to start searching for the "Search for" entry. The options are:

- Beginning of text
- End of text
- Defined position (If selected, the search will start from the character position defined. The value/position must be entered in the adjoining field.)

**For occurrence #:** This number is used to find the nth occurrence of the "Search for" value(s) in the "Parse" entry. e.g.: @OCCUR or 4.

**Search forward (unchecked = backward):** Sets what direction TeleFlow will use to search through the string.

**Case sensitive:** Indicates whether or not a text search should match the case (i.e. lower or UPPER case) or not.

## ACTION STEPS

 **Fail:** The FAIL path is followed when the Parse doesn't find what is indicated in the "Search for" field in the "Parse" entry.

## RELATED STEPS



[Set Global Variable Step](#)



[Set Local Variable](#)



[Compare](#)

### 5.6.1.8.20 Random Number

Generates a random number that falls within a specified range. See the [Variable Usage](#) section for more information on variables.

## PROPERTIES

**Variable to receive random value:** The variable to create or set with the newly generated random number.

**Minimum random value:** The minimum acceptable number to randomly generate and set the variable to.

**Maximum random value:** The maximum acceptable number to randomly generate and set the variable to.

**ACTION STEPS**

None.

**RELATED STEPS**

[Set Global Variable](#)

## 5.6.1.8.21 Read Registry

Reads a value from the registry.

**PROPERTIES**

**Variable to receive value:** The variable to create or modify with the value retrieved from the registry.

**Key name:** The full registry key "path" to seek the registry value in. E.g. HKEY\_LOCAL\_MACHINE \SOFTWARE\TeleFlow

**Value name:** The name of the registry value in the specified registry key to read.

**ACTION STEPS**

**Fail:** Read Registry failed to read the value from the registry key indicated. This may occur for a number of reasons, but the most likely causes are that the user doesn't have permissions to read the registry key indicated, or that the key or value doesn't exist.

## 5.6.1.8.22 Reload Off

Turns off 'safe' reloads for the application. Applications should run Reload Off prior to sections of the application that should not get interrupted. Since [Wait for Call](#) includes a Reload On and Reload Off, inbound/telephone answering applications will always be available for safe shutdowns while idle.

See [Reload On](#) or [TeleFlow Monitor](#) for more information on safe reloads.

**PROPERTIES**

None.

**ACTION STEPS**

None.

**RELATED STEPS**



### [Reload On](#)

#### 5.6.1.8.23 Reload On

Turns on 'safe' reloads for the application. This means that you can restart the application with the latest updated flowchart/TAM information, or perform a safe shutdown from the [TeleFlow Monitor](#) without interrupting active calls.

The [Wait for Call](#) step automatically includes reload on step in it. When the step exits, it includes an automatic reload off step. This means that all inbound/telephone answering applications will allow safe shutdowns while sitting idle. All applications should include reload on and reload off steps (inbound applications already have them, so you need only write them into applications with Wait for Call), placed in appropriate places in the flow that will not cause a safe shutdown or a reload to interrupt important logic, but will allow them to occur.

#### PROPERTIES

None.

#### ACTION STEPS

None.

#### SPECIAL HARDWARE CONSIDERATIONS

Dialogic: The shutdown or reload will occur just after a current call ends, or when a new call is detected. It is best with Dialogic to use "Shutdown all normal" on TeleFlow Monitor when there are active calls, and as soon as they are over, use "Shutdown all abort" in order to avoid busy signals. When all lines are shut down, you can do whatever is was you wanted the reload/shutdown for.

#### RELATED STEPS



### [Reload Off](#)

#### 5.6.1.8.24 Run Basic Script

Runs a script written in TeleFlow's limited implementation of the BASIC programming language.

#### PROPERTIES

**BASIC Script:** The script to run. Each command or operation should go on its own line, but a script can have as many lines as needed to complete the task(s) it is intended to.

The following topics describe the various operations, commands, and functions you can perform and that are available to be used in the Run Basic Script step:

- [AND | OR | XOR | NOT](#)
- [BAND | BOR | BXOR | BNOT | ^](#)
- [CHAR](#)
- [CONCAT](#)

- [DO | WHILE... LOOP \[WHILE | UNTIL\]](#)
- [EXIT](#)
- [FOR ... NEXT \[STEP\]](#)
- [GOSUB...\[EXIT\]...RETURN](#)
- [GOTO](#)
- [IF...THEN...\[ELSEIF\]...\[, ELSE\]...END](#)
- [LEFT](#)
- [LEN](#)
- [LOWERCASE](#)
- [MOD](#)
- [POS](#)
- [PRINT](#)
- [REM | //](#)
- [RIGHT](#)
- [ROUND](#)
- [SELECT CASE](#)
- [Setting TeleFlow variables in Basic Script](#)
- [Standard math operators: + / - \\* =](#)
- [SUBSTR](#)
- [TRIM](#)
- [UPPERCASE](#)

## ACTION STEPS

None.

### 5.6.1.8.24.1 BAND | BOR | BXOR | BNOT | ^

The logical BAND, BOR, BXOR, BNOT bitwise operators and ^ (exponentiate) are logical operators performing element-wise logic:

**BAND:** returns the bitwise AND operator result from both left and right numerical values.

**BOR:** returns the bitwise OR operator result from both left and right numerical values.

**BXOR:** returns the bitwise XOR operator result from both left and right numerical values.

**BNOT:** returns the 32-bit bitwise NOT operator result.

**^:** returns the exponentiate result of the numerical left value exponentiate numerical right value.

### Examples:

<i>BasicScript code</i>	<i>Result</i>
band (7,5)	5
bor (7,5)	7
bxor (7,5)	2
bnot (7)	-8
3^2	9

AND, OR, XOR and NOT are logical operators performing element-wise logic, returning true (nonzero) or false (zero) as follows:

**AND:** if both values on the left and the right side of the operator are different from zero, then AND returns one, otherwise it returns zero.

**OR:** if at least one value from the left or the right side of the operator is different from zero, then OR returns one, otherwise it returns zero.

**XOR:** if one and only one value from the left or the right side of the operator is different from zero, then OR returns one, otherwise it returns zero.

**NOT:** if the value on the right side of this operator is zero, NOT returns one, otherwise it returns zero.

**Examples:**

<b>BasicScript code</b>	<b>Result</b>
print (3 AND 2)	1
print (3 OR 2)	1
print (3 XOR 2)	0
print (3 NOT 2)	0

5.6.1.8.24.2 CHAR

Returns a character based on the ASCII code number in its parameter.

**Syntax:**

CHAR (*number*)

**Examples:**

<b>BasicScript code</b>	<b>Result</b>
@A = CHAR(43) PRINT "@A"	+

**number:** the value associated with the character. Values below 128 are standard and shown here:

VALUE	Character	Description
000	NUL	(Null char.)

001	SOH	(Start of Header)			
002	STX	(Start of Text)			
003	ETX	(End of Text)			
004	EOT	(End of Transmission)			
005	ENQ	(Enquiry)			
006	ACK	(Acknowledgment)			
007	BEL	(Bell)			
008	BS	(Backspace)			
009	HT	(Horizontal Tab)			
010	LF	(Line Feed)			
011	VT	(Vertical Tab)			
012	FF	(Form Feed)			
013	CR	(Carriage Return)			
014	SO	(Shift Out)			
015	SI	(Shift In)			
016	DLE	(Data Link Escape)			
017	DC1 (XON)	(Device Control 1)			
018	DC2	(Device Control 2)			
019	DC3 (XOFF)	(Device Control 3)			
020	DC4	(Device Control 4)			
021	NAK	(Negative Acknowledgement)			
022	SYN	(Synchronous Idle)			
023	ETB	(End of Trans. Block)			
024	CAN	(Cancel)			
025	EM	(End of Medium)			
026	SUB	(Substitute)			
027	ESC	(Escape)			
028	FS	(File Separator)			
029	GS	(Group Separator)			
030	RS	(Request to Send)	031	US	(Unit Separator)
032	SP	(Space)			
033	!	(exclamation mark)			
034	"	(double quote)			
035	#	(number sign)			
036	\$	(dollar sign)			
037	%	(percent)			
038	&	(ampersand)			
039	'	(single quote)			
040	(	(left/opening parenthesis)			
041	)	(right/closing parenthesis)			
042	*	(asterisk)			
043	+	(plus)			
044	,	(comma)			
045	-	(minus or dash)			
046	.	(dot)			
047	/	(forward slash)			
048	0				
049	1				
050	2				
051	3				
052	4				
053	5				
054	6				
055	7				
056	8				
057	9				
058	:	(colon)			
059	;	(semi-colon)			
060	<	(less than)			
061	=	(equal sign)			
062	>	(greater than)			
063	?	(question mark)			
064	@	(AT symbol)			
065	A				
066	B				

067	C	
068	D	
069	E	
070	F	
071	G	
072	H	
073	I	
074	J	
075	K	
076	L	
077	M	
078	N	
079	O	
080	P	
081	Q	
082	R	
083	S	
084	T	
085	U	
086	V	
087	W	
088	X	
089	Y	
090	Z	
091	[	(left/opening bracket)
092	\	(back slash)
093	]	(right/closing bracket)
094	^	(caret/cirumflex)
095	_	(underscore)
096		
097	a	
098	b	
099	c	
100	d	
101	e	
102	f	
103	g	
104	h	
105	i	
106	j	
107	k	
108	l	
109	m	
110	n	
111	o	
112	p	
113	q	
114	r	
115	s	
116	t	
117	u	
118	v	
119	w	
120	x	
121	y	
122	z	
123	{	(left/opening brace)
124		(vertical bar)
125	}	(right/closing brace)
126	~	(tilde)
127	DEL	(delete)

## 5.6.1.8.24.3 CONCAT

Concatenates two strings.

**Syntax:**

```
CONCAT ( string,string)
```

**string:** a string value or variable.

**Examples:**

<i>BasicScript code</i>	<i>Result</i>
<pre>@A = "Hello " @B = CONCAT(@A, "world") PRINT "@B"</pre>	<pre>Hello world</pre>

## 5.6.1.8.24.4 DO | WHILE... LOOP [WHILE | UNTIL]

Executes one or more *statements* in the loop created by DO ... LOOP until certain conditions are met as described by the WHILE or UNTIL statement or until an EXIT DO is reached.

**Syntax:**

```

D O
EXIT DO
LOOP

DO WHILE condition
    [EXIT DO]
LOOP

DO UNTIL condition
    [EXIT DO]
LOOP

D O
    [EXIT DO]
LOOP WHILE condition

D O
    [EXIT DO]
LOOP UNTIL condition
```

**condition:** a BasicScript statement that results in a numeric value. *condition* is said to succeed if the value is not zero and to fail if the value is zero. The WHILE BasicScript keyword will continue a DO loop until *condition* fails (value is zero). The UNTIL BasicScript keyword will continue a DO loop until a *condition* succeeds (value is not zero). If the WHILE or UNTIL keyword is found after the DO, *condition* is evaluated prior to the entry of the DO loop (meaning that there may be no execution of the content of the DO loop based on the result of *condition*). If the WHILE or UNTIL keyword is found after the LOOP, *condition* is evaluated only after the first loop. The WHILE and UNTIL keywords are optional and exclusive. If the DO loop does not contain a WHILE and UNTIL keyword, the EXIT DO is expected to escape the DO loop, otherwise the execution of the code would result in an infinite loop. Note that the syntax of EXIT WHILE and EXIT UNTIL are equivalent to EXIT DO to promote code



readability.

### Examples:

#### **BasicScript code**

#### **Result**

```
PRINT "start"
@A = 1
DO
  PRINT("@A")
  @A = @A + 1
LOOP UNTIL (@A>=5)
PRINT "end"
```

```
start
1
2
3
4
5
end
```

```
PRINT "start"
@A = 10
DO PRINT("@A")
@A = @A + 1
LOOP WHILE (@A<=5)
PRINT "end"
```

```
start
10
end
```

```
PRINT "start"
@A = 10
DO
  PRINT("@A")
  @A = @A + 1
LOOP UNTIL (@A>=5)
PRINT "end"
```

```
start
10
end
```

```
PRINT "start"
@A = 1
DO UNTIL (@A>=5)
  PRINT("@A")
  @A = @A + 1
LOOP
PRINT "end"
```

```
start
1
2
3
4
end
```

```
PRINT "start"
@A = 1
DO WHILE (@A<=5)
  PRINT("@A")
  @A = @A + 1
LOOP
PRINT "end"
```

```
start
1
2
3
4
5
end
```

```
PRINT "start"
@A = 10
DO WHILE (@A<=5)
  PRINT("@A")
  @A = @A + 1
LOOP
PRINT "end"
```

```
start
end
```

```

PRINT "start"           start
@A = 10                 end
DO UNTIL (@A>=5)
  PRINT("@A")
  @A = @A + 1
LOOP
PRINT "end"

PRINT "start"           start
@A = 1                 1
DO                     2
  PRINT("@A")           3
  @A = @A + 1           4
  IF @A >= 5 THEN      5
    EXIT DO             end
  END
LOOP
PRINT "end"

```

#### 5.6.1.8.24.5 EXIT

Immediately exits the script.

#### Syntax:

```
EXIT
```

#### Example:

<i>BasicScript code</i>	<i>Result</i>
@A = "Hello world" Print "@A" EXIT PRINT "Hello again"	Hello world

#### 5.6.1.8.24.6 FOR ... NEXT [STEP]

Executes one or more *statements* in the FOR ... NEXT loop a specified number of times as defined by the range of *start* and *end* values as incremented by the optional STEP *stepvalue* (default value is 1). The EXIT FOR (optional) keyword in a FOR loop forces a premature exit of the loop and resumes with the remainder of the BasicScript.

#### Syntax:

```

FOR variable = start TO end [STEP stepvalue]
  BasicScript statement (s)
[EXIT FOR]
  BasicScript statement (s)
NEXT

```

**variable:** variable name to increment on each NEXT keyword. TeleFlow (including BasicScript) variables are always preceded by a '@' character.

**start:** beginning value (integer) at which the variable starts the FOR loop.  
**end:** final value (integer) at which the variable stops (and exits) the FOR loop.  
**stepvalue:** the increment on each NEXT of variable. An integer value is expected. This value can't be zero. The STEP value can be omitted, in which case the *stepvalue* is one.

Examples:

<b>BasicScript code</b>	<b>Result</b>
PRINT "start"	Start
FOR @VAR = 1 TO 10 STEP 2	1
PRINT "@VAR"	3
NEXT	5
PRINT "end"	7
	9
	end
PRINT "start"	Start
FOR @VAR = 5 TO 1 STEP -1	5
PRINT "@VAR"	4
NEXT	3
PRINT "end"	2
	1
	end
PRINT "start"	start
FOR @VAR = 1 TO 5	1
PRINT "@VAR"	2
NEXT	3
PRINT "end"	4
	5
	end
PRINT "start"	start
FOR @VAR = 1 TO 100	1
PRINT "@VAR"	2
IF @VAR >= 5 THEN	3
EXIT FOR	4
END	5
NEXT	end
PRINT "end"	
PRINT "start"	start
VARDDATE = "2005/01/01"	2005/01/01
FOR @VAR = 1 TO 5	2005/01/02
@TOTVARDDATE = @VARDDATE + @VAR	end
IF @TOTVARDDATE >= "2005/01/03	
THEN	
EXIT FOR	
PRINT "@TOTVARDDATE"	
END	
NEXT	
PRINT "end"	

#### 5.6.1.8.24.7 GOSUB ... RETURN

The GOSUB keyword is used to create subroutines. The RETURN keyword is used to return from the subroutine to the command following the GOSUB that called the subroutine.

**Syntax:**

```
GOSUB subroutine
BasicScript statement (s)
subroutine:
RETURN
```

***subroutine***: name chosen for the subroutine.

Note that within the subroutine code, variables are referred to and affected as global variables. Exclusively global variables are available in this version of BasicScript.

### Examples:

<i>BasicScript code</i>	<i>Result</i>
<pre>PRINT "start" GOSUB doitonce GOSUB doitonce</pre>	<pre>start Hello Hello</pre>
<pre>doitonce:   PRINT "Hello" RETURN</pre>	
<pre>PRINT "end" GOTO final</pre>	<pre>end</pre>
<pre>doitonce:   PRINT "Hello" RETURN</pre>	
<pre>final:</pre>	

#### 5.6.1.8.24.8 GOTO

Jumps to a label statement.

### Syntax:

```
GOTO label
label:
```

### Examples:

<i>BasicScript code</i>	<i>Result</i>
<pre>@VARDATE = @DATE + 1 GOTO final PRINT "end"</pre>	<pre>2005/01/02</pre>
<pre>final:   PRINT @VARDATE</pre>	

## 5.6.1.8.24.9 IF...THEN...[ELSEIF]...[, ELSE]...END

Use the IF THEN ELSE END commands to perform conditionally a sequence of BasicScript statements.

**Syntax:**

```
IF condition THEN
[ELSEIF condition]
[ELSE]
END
```

**condition:** BasicScript commands or variable that result in a numeric value. If the value is zero, *condition* is said to have failed. If the value is not zero, *condition* is said to have succeeded. The IF keyword is followed by a condition statement, and the THEN keyword. ELSEIF is used like a concatenation of ELSE and another IF and does not require its own END keyword.

**Examples:**

<b>BasicScript code</b>	<b>Result</b>
<pre>PRINT "start" @A = 1 IF @A = 1 THEN   PRINT "Hello" ELSEIF @A = 2 THEN   PRINT "WORLD" ELSE   PRINT "Huh" END PRINT "end"</pre>	<pre>start Hello end</pre>
<pre>PRINT "start" @A = 2 IF @A = 1 THEN   PRINT "Hello" ELSEIF @A = 2 THEN   PRINT "WORLD" ELSE   PRINT "Huh" END PRINT "end"</pre>	<pre>start World end</pre>
<pre>PRINT "start" @A = 3 IF @A = 1 THEN   PRINT "Hello" ELSEIF @A = 2 THEN   PRINT "WORLD" ELSE   PRINT "Huh" END PRINT "end"</pre>	<pre>start Huh end</pre>

```
PRINT "start"           start
@A = 3                 end
IF @A = 1 THEN
  PRINT "Hello"
ELSEIF @A = 2 THEN
  PRINT "WORLD"
END
PRINT "end"
```

#### 5.6.1.8.24.10 LEFT

Returns a specified number of leftmost characters in a string.

#### Syntax:

```
LEFT (string, number)
```

***string***: a string value or variable.

***number***: the number of leftmost characters to return from *string*.

#### Examples:

<b><i>BasicScript code</i></b>	<b><i>Result</i></b>
@A = "Hello world" @B = LEFT(@A, 5) PRINT "@B"	Hello

#### 5.6.1.8.24.11 LEN

Returns the number of alphas in a string.

#### Syntax:

```
LEN (string)
```

***string***: a string value or variable.

#### Examples:

<b><i>BasicScript code</i></b>	<b><i>Result</i></b>
@A = "Hello world" @B = LEN(@A) PRINT "@B"	11

## 5.6.1.8.24.12 LOWERCASE

Returns a string's lowercase equivalent.

**Syntax:**

LOWERCASE (*string*)

**string:** a string value or variable.

**Examples:**

<i>BasicScript code</i>	<i>Result</i>
@A = LOWERCASE("HELLO WORLD") PRINT "@A"	hello world

## 5.6.1.8.24.13 MOD

Divides two integer expressions, returning only the remainder.

**Syntax:**

*dividend* MOD *divisor*

**dividend:** integer divided into

**divisor:** integer divided by

**Example:**

<i>BasicScript code</i>	<i>Result</i>
print (3 MOD 2)	1

## 5.6.1.8.24.14 POS

Returns the indexed position of a substring in a string, or -1 if the substring is not part of a string.

**Syntax:**

POS (*string*, *substring*)

**string:** a string value or variable.

**substring:** the substring to search for in *string*.

**Examples:**

<i>BasicScript code</i>	<i>Result</i>
@A = "HELLO WORLD" @B = POS (@A, "WORLD") PRINT "@B"	6
@A = "HELLO WORLD" @B = POS (@A, "AGAIN") PRINT "@B"	-1

## 5.6.1.8.24.15 PRINT

Prints string's content to the output.

**Syntax:**

```
PRINT (string)
```

***string***: a string value or variable.

**Examples:**

<i>BasicScript code</i>	<i>Result</i>
@A = "Hello world" PRINT "@A" PRINT "Hello again"	Hello world Hello again

## 5.6.1.8.24.16 REM | //

Holds non-executing comments.

**Examples:**

```
REM This entire line is a comment.  
// This entire line is also a comment.
```

## 5.6.1.8.24.17 RIGHT

Returns a specified number of rightmost characters in a string.

**Syntax:**

```
RIGHT (string, number)
```

***string***: a string value or variable.

***number***: the number of rightmost characters to return from *string*.

**Examples:**



<i>BasicScript code</i>	<i>Result</i>
@A = "Hello world" @B = RIGHT(@A,5) PRINT "@B"	world

## 5.6.1.8.24.18 ROUND

Returns the nearest integer value to the numeric value parameter

**Syntax:**

ROUND (*number*)

***number***: a number value.

**Examples:**

<i>BasicScript code</i>	<i>Result</i>
@A = ROUND(1.25) PRINT "@A"	1 2
@A = ROUND(1.90) PRINT "@A"	

## 5.6.1.8.24.19 SELECT CASE

Use the SELECT CASE BasicScript keywords to test variable values (numeric or alpha).

**Syntax:**

```
SELECT CASE variable
  CASE value [TO value][value]
  [CASE value [TO value][value]]
  [CASE ELSE]
END SELECT
```

***variable***: variable name to test in each CASE statement against. TeleFlow (including BasicScript) variables are preceded by a '@' character.

***value***: a numeric or alpha value to test variable the against.

Use the TO keyword to test value ranges in a CASE statement. The TO statement expects a numeric or a single letter string value. Use CASE ELSE to perform some operations if all preceding CASE failed.

**Examples:**

<i>BasicScript code</i>	<i>Result</i>
<pre>PRINT "start" @A = 1 SELECT CASE @A   CASE 2 TO 5     PRINT "Hello"   CASE 1     PRINT "World"   CASE ELSE     PRINT "Huh" END SELECT PRINT "end"</pre>	<pre>start World End</pre>
<pre>PRINT "start" @A = 4 SELECT CASE @A   CASE 2 TO 5     PRINT "Hello"   CASE 1     PRINT "World"   CASE ELSE     PRINT "Huh" END SELECT PRINT "end"</pre>	<pre>start Hello End</pre>
<pre>PRINT "start" @A = "C" SELECT CASE @A   CASE "A" TO "F"     PRINT "Hello"   CASE "Z"     PRINT "World"   CASE ELSE     PRINT "Huh" END SELECT PRINT "end"</pre>	<pre>start Hello End</pre>
<pre>PRINT "start" @A = "Y" SELECT CASE @A   CASE "A" TO "F"     PRINT "Hello"   CASE "Z"     PRINT "World"   CASE ELSE     PRINT "Huh" END SELECT PRINT "end"</pre>	<pre>start Huh end</pre>
<pre>PRINT "start" @A = "Y" SELECT CASE @A   CASE "A" TO "F"     PRINT "Hello"   CASE "Z"     PRINT "World" END SELECT PRINT "end"</pre>	<pre>start end</pre>

## 5.6.1.8.24.20 Setting TeleFlow variables in Basic Script

Setting TeleFlow variables in the Run Basic Script step is accomplished by simply assigning a value to a TeleFlow variable with an '=' sign.

For example, the following line will set (or create, if it doesn't already exist) the [global variable](#) @MY\_VARIABLE to 5:

```
@MY_VARIABLE = 3 + 2
```

To set or create a local variable using Run Basic Script, use the "LOCAL" tag just prior to the variable name. For example, the following line will create/set local variable @MY\_LOCAL\_VARIABLE to 6:

```
LOCAL @MY_LOCAL_VARIABLE = 3 + 3
```

## 5.6.1.8.24.21 Standard math operators: + / - \* =

Performing mathematical calculations with the Run Basic Script step only requires the use of the basic math operators. They are:

- =: Set the variable on the left side of the = side equal to the total calculated value of the operations on the right. See examples with the operations below.
- +: Add (Eg. "@MY\_VARIABLE = 5 + 5" sets TeleFlow variable @MY\_VARIABLE to 10)
- /: Divide (Eg. "@MY\_VARIABLE = 5 / 5" sets TeleFlow variable @MY\_VARIABLE to 1)
- -: Subtract (Eg. "@MY\_VARIABLE = 5 - 5" sets TeleFlow variable @MY\_VARIABLE to 0)
- \*: Multiply (Eg. "@MY\_VARIABLE = 5 \* 5" sets TeleFlow variable @MY\_VARIABLE to 25)
- ^: Exponent (Eg. "@MY\_VARIABLE = 5 ^ 5" sets TeleFlow variable @MY\_VARIABLE to 3125)
- (): Brackets groups operations to ensure certain calculations take precedence, per "standard math"(brackets, exponents, division, multiplication, addition, subtraction). Examples:
  - "@MY\_VARIABLE = 5 \* 5 + 5 \* 5" sets TeleFlow variable @MY\_VARIABLE to 50
  - "@MY\_VARIABLE = (5 \* 5 + 5) \* 5" sets TeleFlow variable @MY\_VARIABLE to 150

## 5.6.1.8.24.22 SUBSTR

Extracts a substring from a string value.

**Syntax:**

```
SUBSTR (string, number1, number2)
```

**string:** a string value or variable.

**position:** the starting position (index) in the string. Must be numeric, valid range is from 1 (the first character in the string) to the total number of characters in the string.

**IMPORTANT NOTE:**

Prior to TeleFlow build number 2007.1123, the Run BASIC Script's SUBSTR function was 0-based (0 being the first character in the string).

**numberchars:** the number of characters to copy.

**Examples:**

<i>BasicScript code</i>	<i>Result</i>
@A = SUBSTR("HELLO WORLD",7,2) PRINT "@A"	WO

## 5.6.1.8.24.23 TRIM

Trims space characters from both ends of a string.

**Syntax:**

TRIM (*string*)

*string*: a string value or variable.

**Examples:****BasicScript code**

```
@A = "  HELLO WORLD  "
@B = TRIM(@A)
PRINT "@A"
PRINT "@B"
```

**Result**

```
HELLO WORLD
HELLO WORLD
```

## 5.6.1.8.24.24 UPPERCASE

Returns a string's uppercase equivalent.

**Syntax:**

UPPERCASE (*string*)

*string*: an string value or a variable.

**Examples:****BasicScript code**


```
@A = UPPERCASE("hello world")
PRINT "@A"
```


**Result**

```
HELLO WORLD
```

## 5.6.1.8.25 Run Flowchart

Runs a TeleFlow Flowchart.

When a flowchart is run, the application flow enters the flowchart, starts running at the  Start

step, and continues until it reaches the  End step. When the flow reaches the End step, the flowchart ends/exits, and the application continues from the Run Flowchart step in the flowchart that ran it. Every flowchart is a file on disk that is named with a .TAM (Telephony Application Module) extension. This essentially describes how a TeleFlow application works: Flowcharts are developed using TeleFlow steps to create application/call flow. Each flowchart performs a task or series of tasks. Stringing together series of flowcharts that run other flowcharts creates an overall application/call flow.

## PROPERTIES

**Flowchart:** The name of the flowchart to run. Any flowchart run must be added to the TeleFlow Application (typically, adding a flowchart to an application is the first step of developing it, and as such, most application flowcharts are properly added as soon as they are created). Flowcharts are added to the application on the TeleFlow Application (.TAP) screen. e.g.: GetAcct

**Pass parameters to Flowchart (Value 1 - 10):** A Flowchart can receive up to 10 pieces of data from a Run Flowchart step that runs it, receiving each into a variable. On the Run Flowchart step, this is represented by numbered "Value" fields. If a parameter on Run Flowchart is left empty, but the Flowchart is set to receive a parameter into a variable, the variable is created/set to an empty string.

## ACTION STEPS

None.

## RELATED STEPS

[Run TAP](#)



[Go to Label](#)

### 5.6.1.8.26 Run Program

Runs an external program or batch file and captures the results into the variable specified. Programs can be an executable (.EXE), command (.CMD), batch (.BAT) or any other file type which can be executed from within Windows.

## PROPERTIES

**External program to run (EXE):** A valid directory path and filename of the program to execute/run. e.g.: c:\ivrapp\bin\SendEmail.bat

**Variable to receive result:** If the program executed returns a result, it is stored in the variable defined here. e.g.: @Result

## ACTION STEPS

None.

## RELATED STEPS



[Run Flowchart](#)

### 5.6.1.8.27 Run TAP

Runs another TeleFlow Application, or TAP file.

Special care must be taken when developing a TeleFlow Application that is expected to run from within another application. The sub-application (I.e. the one to run) must be designed such that it will not run a Wait for Call step if a caller is already on the phone, and will exit when someone hangs up or when an error occurs (and when whatever task the application was run for is complete, of course!).

## PROPERTIES

**TeleFlow Application (.TAP):** The TeleFlow Application to run.

**Pass through database connections:** Provides the current application's database connection to the application that is being run. The application run can run queries against that database connection without running the [DB Connect](#) step to attain its own connection (I.e. It is effectively the same as the application being run having attained the database connection using DB Connect). This option is provided for convenience, and because some database connections can be slow to connect, which means that database connections while a caller is on the line are undesirable.

**Parameter/Value (@P1 - @P9):** The data to pass in to the application being run. The application will receive these as the application parameter variables @P1 through @P9. This works just like the parameter settings in [TeleFlow Line List](#), wherein the parameters for each line for variables @P1 through @P9 are defined prior to running TeleFlow Server. **SPECIAL:** The START step in the TAP screen of the application receiving variables passed in by a Run TAP step can also define new global variables to receive the values passed in. These can even be used in a pass by reference fashion to return information to the application using the Run TAP step. See [Starting Point](#) and the description on [Flowchart Parameters](#) for more information: Passing application parameter variables by value and reference work the same way for TeleFlow Applications as they do for flowcharts.

## BUTTONS

**Locate:** Opens a file location dialog window. Use this button to find a the TAP file to run.

## ACTION STEPS

None.

## RELATED STEPS



[Run Flowchart](#)

### 5.6.1.8.28 Save Log

Saves the contents of the TeleFlow application log to a file on disk.

## PROPERTIES

**Save to TeleFlow Log folder:** This option uses the current TeleFlow Log directory setting as the

location to save the log output. See [TeleFlow Config](#) for more information on setting the TeleFlow log directory.

**Save to folder:** This options saves the log to the directory specified as long as it is a valid Windows path. The value provided has no bearing on the current TeleFlow log directory setting in [TeleFlow Config](#). The setting takes affect for this step only.

**Use TeleFlow Log name:** This option uses the name associated with TeleFlow Application (TAP) as the file name to write the log information to. The name is defined when creating a list of applications to run in [TeleFlow LineList](#).

**Specify file name:** This option specifies the name of the file to log to. Provide a valid Windows filename.

## ACTION STEPS

None.

## RELATED STEPS



[Write Text](#)

### 5.6.1.8.29 Scan Folder

Gather a list of all files in a directory and/or a count of files in a directory.



If desired, the [Parse step](#) can be used to read each filename one at a time (in a programmatic loop) from the variable containing the file list. Each file in the list is separated by a linefeed character. The [@LINEFEED](#) system variable contains a single linefeed character, which you can use in the "Search for Text String" field of the Parse step.

## PROPERTIES

**Scan directory (folder):** A valid directory path to find files in. e.g.: C:\VoiceMail\NewMessages or @MessagesDirectory

**With file mask (eg: \*.txt):** A file mask to limit the file names returned by the step. Supports wildcard substitution (I.e. Use ? to indicate any single character, and \* to indicate any number of different characters). To return all file names, use \*.\*.

**Variable to receive directory list:** The variable to receive the directory file list into.

**Variable to receive directory count:** The variable to receive the total number of files located in the directory.

## ACTION STEPS



**Fail:** The directory scan failed. This may occur for a number of reasons, but the most likely causes are that directory name was not valid, or the directory does not exist.

## RELATED STEPS



[Create Folder](#)



[Parse](#)

### 5.6.1.8.30 Set Global Variable

Creates and/or sets a global variable. A global variable in TeleFlow is available throughout the application flow from the point at which it is created. The format definition of the variable will determine what type of information the variable represents. For example, '2001/01/22' will represent a date, whereas 'NORTH AMERICA' represents a character value. See the [Variable Usage](#) section for more information on variables, variable scope, and variable data types..

## PROPERTIES

**Variable to define:** The name of the variable to define e.g.: @VarName. Variables must be named according to the following conventions:

1. All TeleFlow variables start with the @ symbol.
2. Only letters (a...z, A...Z), numbers (0...9) and underscores ( \_ ) can be used in variable names.
3. While not required, the TeleFlow standard when typing in variables is that they are all named in ALL CAPS, for purposes of readability.
4. For variable names, TeleFlow is not case sensitive. It treats all variables as being in UPPER CASE, regardless of how they are referred to in the application.

**Assigned value:** The value to assign to the variable defined.

## ACTION STEPS

None.

## RELATED STEPS



[Set Local Variable](#)



[Math](#)

### 5.6.1.8.31 Set Language

Sets the language to use and the audio file path for the [Say](#) step.

This step is the only mechanism for changing where the Say step audio files are located.

If this step is not used, the Say step will use English by default.

## PROPERTIES

**Language Selection:** The language selection sets the rules for how numbers, times, and dates are spoken by the Say step. 3 language modules are provided by default: English, French, and Spanish. Custom modules can be loaded by selecting -- Custom -- from the pull-down list.



**Language Rules program:** Specifies the custom language module when -- Custom -- is selected from the Language Selection pull-down menu. For information on creating a custom language module, contact enGenic.

**Location of prompt recordings:** Path specifying the directory where the audio files for Say step are found. This can set to use custom prompts for any language module including the defaults. If this field is left blank, the default audio file locations from the TeleFlow install will be used. e.g. <TeleFlow install directory>\Languages\English. See the topic on [Voice Prompts](#) for a listing of the prompt files that must be recorded in order to supply custom prompt recordings for an application.

## BUTTONS

**Locate:** Opens a file location dialog window. Use this button to find a language rules program.

## RELATED STEPS



[Say Step](#)

### 5.6.1.8.32 Set Local Variable

Creates and/or sets a local variable. A Local Variable is the same as a Global Variable, except that the value set will only exist until the application exits the current flowchart. The format definition of the variable will determine what type of information the variable represents. For example, '2001/01/22' will represent a date, whereas 'NORTH AMERICA' represents a character value. See the [Variable Usage](#) section for more information on variables, variable scope, and variable data types.

## PROPERTIES

**Variable to define:** The name of the variable to define e.g.: @VarName. Variables must be named according to the following conventions:

1. All TeleFlow variables start with the @ symbol.
2. Only letters (a...z, A...Z), numbers (0...9) and underscores ( \_ ) can be used in variable names.
3. While not required, the TeleFlow standard when typing in variables is that they are all named in ALL CAPS, for purposes of readability.
4. For variable names, TeleFlow is not case sensitive. It treats all variables as being in UPPER CASE, regardless of how they are referred to in the application.

**Assigned value:** The value to assign to the variable defined.

## ACTION STEPS

None.

## RELATED STEPS



[Set Global Variable](#)



[Math](#)

## 5.6.1.8.33 Set Play Variable

Defines or sets a variable to a Script code (see the [Voice Script Manager](#) for more information on Script codes). The intention of this step is enhance the readability of a TeleFlow application. While you could set a variable to a Script code normally, and then use a Play step to Play the variable, doing so with Set Play Var is preferable. This is because the Set Play Var step allows you to see the script for the voice message code where you set the Script code variable. See the [Variable Usage](#) section for more information on variables.

**PROPERTIES**

**Variable to define:** The TeleFlow variable to populate with the Script code.

**Script code:** The Script code assigned to the Variable to define. Any documentation that was done in the Voice Script Manager will duplicate itself in the notes for the Set Play Variable step. E.g.: MAIN\_MENU01 or @VOICECD

**BUTTONS**

**Voice Manager:** A short cut to the [Voice Script Manager](#). The Voice Script Manager is also accessible from the "Window" menu of TeleFlow Designer.

**Play:** Plays the voice file associated with the Script code, if the file is accessible in design time.

**ACTION STEPS**

None.

**RELATED STEPS**

[Play](#)



[Set Global Variable](#)

## 5.6.1.8.34 Wait

Pauses the application for a definable amount of time. Use this command to place breaks between speech, or to idle the application temporarily.



Another method to create a wait is to use the [Play](#) step to play a recording of silence.

**PROPERTIES**

**Delay (ms):** The time, in milliseconds, to wait. e.g.: @WaitTime or 500

**Interruptible:** If checked, this step will be skipped (i.e. no wait will occur) if there are pending tones in the [touch-tone buffer](#), or interrupted if a key is pressed during the pause. If the caller speaks and the system has barge-in enabled for Speech Recognition, this will also skip/interrupt the Wait. If it is not interruptible, a caller cannot interrupt or skip the Wait.

## ACTION STEPS

None.

## RELATED STEPS



[Play](#)

### 5.6.1.8.35 Write Text

Outputs text to the TeleFlow log or a text file.

## PROPERTIES

**Message text:** The text to output. This can be a combination of standard text and variables. e.g.:  
Current account: @AcctNo Balance: @AcctBal

**Start new line after message text:** If checked, a carriage return is included at the end of the text. This means that if/when another line of text is output to the same file, it will start on a new line. In order to construct one message line with several message commands, do not check the start new line box.

**File output:** If defined, the message text output will be directed to a file. Otherwise, the Message text is output to the TeleFlow log. e.g.:c:\ivrapp\txt\textfile.txt or @TextFile

**Append:** If checked, the message text will append to the file if it exists, or create a new file if it doesn't. If unchecked, the file will be overwritten if it exists, and created if it does not.

## ACTION STEPS

None.

## RELATED STEPS



[Load Text](#)

### 5.6.1.9 Advanced Steps (Adv. tab)

[Set HW Parameter](#)

#### 5.6.1.9.1 Set HW Parameter

Sets a value to a particular hardware parameter or setting.

It is advised that hardware settings **only be changed by advanced users**. Consult the documentation appropriate for your hardware for more information on the hardware parameters.

## PROPERTIES

**Name of hardware setting:** Supply the name of the hardware setting or parameter you wish to change. Some examples follow.

On NMS hardware, SR barge-in sensitivity can be adjusted with the following hardware settings:

EVAD.SNR	signal to noise ratio (5 to 30 dB)
EVAD.MIN_NOISE	minimum noise floor (-100 to -40 dB)
EVAD.MAX_NOISE	maximum noise floor (-65 to -25 dB)
EVAD.	signal attack time constant (10 to 200 ms)
SIGNAL_ATTACK	

By default, the SR barge-in sensitivity is set to optimal values.  
These should only be adjusted if you are experiencing problems with barge-in.

**Value:** The value you wish to assign to the hardware parameter or setting. In nearly all cases, this value must be an integer (or evaluate to an integer if you supply a TeleFlow variable).

## ACTION STEPS



Fail: The parameter or setting could not be set to the value specified.

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

Ai-Logix  
NMS Communications

## RELATED STEPS



[SR Barge-In On](#)

### 5.6.1.10 SR

The following list is comprised of all of the Speech Recognition (and some loosely related) steps that are available in TeleFlow Designer. These steps are found under the SR tab of the [Steps Toolbox Window](#).

[Menu](#)  
[Speak Text](#)  
[SR Acquire](#)  
[SR Barge-In On](#)  
[SR Barge-In Off](#)  
[SR Clear Voice Buffer](#)  
[SR Flag Result](#)  
[SR Get Digits](#)  
[SR Get String](#)  
[SR Get Yes/No](#)  
[SR Release](#)  
[SR VoicePass Auth](#)  
[SR VoicePass Reg](#)

## Speech Recognition (SR) Applications

SR applications require a speech recognition provider, such as [LumenVox](#) or [Nuance](#), and SR capable hardware. Currently, TeleFlow barge-in speech recognition is only supported on NMS, however, prompted speech recognition is supported on both Dialogic and NMS. Support for active applications, including barge-in speech recognition, on Ai-Logix is under development.

Although for prompted speech recognition applications, the SR Acquire step can occur as needed, for instance, for dynamic resource usage, barge-in speech recognition applications require that the SR Acquire step occur before the call is processed (such as the Wait for Call step). Barge-in speech recognition applications can only acquire speech recognition resources outside of a call.

If SR steps, other than SR Acquire and SR Release, occur before an SR Acquire step, the application will terminate with a FATAL error.

During any speech recognition step, touch tones have precedence over speech. For the SR Get Digits, SR Get Yes/No, and SR Get String steps, if a touch tone is pressed during speech, the speech audio will be ignored, and the step will process the THRESHOLD action step. For the Menu step, if a touch tone is pressed during speech, the speech audio will be ignored, and the step will process the associated action step.

**Note:** Speech Recognition is not available as a part of the standard TeleFlow Server. You must purchase additional software to use SR functions.

### See also:






- [LumenVox Speech Recognition](#)
- [Nuance Speech Recognition](#)

#### 5.6.1.10.1 Menu

The Menu step is available for both touch-tone and speech recognition input.

Collects a selection through a touch tone or speech recognition, following the path related to the user's selection.


Speech recognition is only available if an [SR Acquire](#) step was previously executed. Without SR, this step will accept Touch-Tone input only.

When this step is placed on the flowchart, most of its related Action steps will also be created. Any Action steps that are not required can be hidden through an option on the right mouse button menu, or by pressing Ctrl-H on each. If the user should enter an option that is hidden, the Server will follow the path of the  All Others Action step. The four Action steps for the letters,    and , are automatically hidden. To use any of these four action steps, you must unhide them from the step's right mouse button menu.

## PROPERTIES

**Variable to receive selection:** The touch tone digit or speech recognition result will be sent to this TeleFlow variable. This field is blank by default, but may be any TeleFlow variable. Note that no result variable needs to be collected by this step; collecting a result variable with this step is optional.

**TIMEOUT (ms):** This field is the maximum time, in milliseconds, before this step will time out and

follow  [Time Out](#) path if no touch tone or speech entry is detected. If this field is blank (the default), [@TIMEOUT](#) will be used.

**Speech Recognition Options:** These options are only necessary/used when speech recognition is enabled. If they are all left blank in a speech recognition enabled system, the step reverts to using touch-tone only behavior.

**THRESHOLD confidence %:** The result confidence is compared against this minimum confidence to determine a valid result. This field is populated with 50 by default. This field must not be blank.

**Variable to receive confidence %:** The result confidence will be put in this TeleFlow variable. This field is populated with [@SR\\_CONFIDENCE](#) by default. This field can be blank.

**Remember this recognition for verification through SpeechTrainer:** If the [TeleFlow SpeechTrainer](#) is active, checking this box ensures the recognition will be available for review in SpeechTrainer.








**Variable to receive SR Handle:** Use this to capture the SR Handle for the utterance if the application uses the [SR Flag Result](#) step to flag certain types of SR results in SpeechTrainer. Utterances flagged in SpeechTrainer can be found more easily, and used to isolate specific problems the application is set up to identify.


**Retain audio on THRESHOLD:** Check this checkbox to compare the speech recognition audio against another grammar (such as SR Get Yes/No or SR Get String) when the result confidence is less than the THRESHOLD confidence.

**Base grammar (name or file):** This field specifies the name of the Nuance pre-compiled grammar, or the SRGS grammar file for use with LumenVox. If this field is left blank, only the additional grammar text will be used for speech recognition. If both this field and the additional grammar field are left blank, this step will respond to touch tones only.


**Additional grammar text:** This field specifies SRGS grammar rules in addition to the base grammar, but can only be used by LumenVox. If this field is left blank, only the base grammar will be used for speech recognition. If both this field and the additional grammar field are left blank, this step will respond to touch tones only.

## ACTION STEPS

 through , , ,  through  : A touch tone was detected, and the confidence was set to 100, **or** the speech recognition grammar returned a result matching the number indicated with adequate confidence. (E.g. The  path would be followed if a caller presses "0", or if the result from Speech Recognition was "0" and the confidence was equal to or higher than the THRESHOLD confidence %)

 **All Others:** A touch tone was detected that did not correspond to a connected action step, **or** a speech recognition result was detected with adequate confidence that did not correspond to a connected step. Use the variable to receive selection to determine the result.

 [Unrecognized Voice Action:](#) The confidence was less than the THRESHOLD confidence.

 [Time Out:](#) No speech or touch tones were detected before the No-Voice TIMEOUT elapsed.

## SPECIAL HARDWARE CONSIDERATIONS

- Dialogic: Only touch-tone and prompted SR are supported (I.e. SR with barge-in is not currently supported).
- Ai-Logix: Touch-tone support only.

## RELATED STEPS



[Get TT String](#)



[Get Letter](#)



[SR Acquire](#)



[SR Get Yes/No](#)



[SR Get String](#)



[SR Get Digits](#)

### 5.6.1.10.2 Speak Text

This step converts text to speech, playing the resulting audio to the caller.

Note: Text-to-Speech is not available as a part of the standard TeleFlow Suite. You must purchase additional software to use this step.

## PROPERTIES


**Server IP address for TTS Server computer:** The IP address of the computer hosting the TTS Server. This must be an IP address and not a computer name or URL. This field must be filled in even if the local computer is the one hosting the TTS Server.

**IP port:** IP port for the TTS Server. 7000 is the default.

**Text to speak:** Supply the text the step must convert and speak. The text can be a TeleFlow variable, or a combination of literal text and variables. E.g. "Hello @FIRST\_NAME, this is @COMPANY\_NAME calling."

**Interruptible:** If checked, this step will be skipped (I.e. no tone will be played) if there are pending tones in the [touch-tone buffer](#), or interrupted if a key is pressed while the step is playing audio. If the caller speaks and the system has barge-in enabled for Speech Recognition, this will also skip/interrupt the audio. If it is not interruptible, the user cannot interrupt or skip the step.

## ACTION STEPS

 **Failure:** A failure will occur if the TTS Server does not respond, which can happen for a number of reasons. The three most common are the IP address supplied is incorrect, the IP port supplied is incorrect, or a TTS Server is not running on the computer specified by the IP address.

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

Ai-Logix  
Dialogic  
NMS Communications

### 5.6.1.10.3 SR Acquire

Acquires resources for speech recognition. This step must be used before any of the other SR steps will function.

## PROPERTIES


**Speech Recognition Module:** Choose Voice\_LumenVox.dll for LumenVox 6.x or Voice\_Nuance.dll for Nuance 7.04. This field must not be blank.

**Speech Recognition Parameters:** Enter the path and filename of the parameter file for LumenVox, or the path to the directory of compiled grammars for Nuance. This field must not be blank.

**Speech Recognition end of speech timeout (ms):** Enter the silence time, in milliseconds, after a speaker has finished speaking, to mark the end of a speech recognition audio segment. If left blank, the current value is used. The default value is 750 ms.

**Enable Speech Recognition Barge-In:** In order to use barge-in with speech recognition, this checkbox must be checked and the resource must be acquired before a call begins. After the resource is acquired, barge-in is active and barge-in may be turned on and off with the Barge-In On and Barge-In Off steps.

## ACTION STEPS

 **Fail:** The SR resource could not be acquired, possibly because there are inadequate licenses.

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

Dialogic (prompted only)  
NMS Communications

## RELATED STEPS



[SR Release](#)



[SR Barge-In On](#)





### [SR Barge-In Off](#)

#### 5.6.1.10.4 SR Barge-In Off

De-activates speech recognition barge-in(see [SR Acquire](#) and [SR Barge-In On](#)). If barge-in is not active, this step has no effect.

#### PROPERTIES

None.

#### ACTION STEPS

None.

#### SUPPORTED TELEPHONY HARDWARE PLATFORMS

NMS Communications

#### RELATED STEPS



### [SR Barge-In On](#)



### [SR Acquire](#)



### [SR Release](#)

#### 5.6.1.10.5 SR Barge-In On

Activates speech recognition barge-in if "Enable Speech Recognition Barge-In" was checked in [SR Acquire](#).

If SR Barge-In resources have not been acquired by an SR Acquire step, the application will terminate with a FATAL error. If barge-in is already active, this step has no effect.

#### PROPERTIES

None.

#### ACTION STEPS

None.

#### SUPPORTED TELEPHONY HARDWARE PLATFORMS

NMS Communications

**RELATED STEPS**[SR Barge-In Off](#)[SR Acquire](#)[SR Release](#)

## 5.6.1.10.6 SR Clear Voice Buffer

Clears speech recognition audio retained by a previous SR step for a THRESHOLD result. While speech recognition audio is pending processing by an appropriate SR step (such as [SR Get String](#)), interruptible steps (i.e. Steps with a checkbox labeled "Interruptible") will be interrupted immediately by TeleFlow. SR Clear Voice Buffer clears such pending audio.

**PROPERTIES**

None.

**ACTION STEPS**

None.

**SUPPORTED TELEPHONY HARDWARE PLATFORMS**

Diallogic  
NMS Communications

**RELATED STEPS**[SR Get Digits](#)[SR Get Yes/No](#)[SR Get String](#)

## 5.6.1.10.7 SR Flag Result

Flags a speech recognition result for examination with the [TeleFlow SpeechTrainer](#). When reviewing speech recognition results in SpeechTrainer, results can be filtered to show only flagged results, making it easier to find those that the application developer is specifically trying to locate.

'Save SR Audio to Log Directory' must be checked in [TeleFlow Config](#) for the appropriate Speech Trainer files to be generated and flagged.

## PROPERTIES

**SR Handle of previously remembered recognition:** The variable containing the SR handle from the SR step of the recognition to flag. An SR handle can be gathered from any step that performs speech recognition(The step must check the "Remember this recognition for verification through SpeechTrainer" checkbox, and must supply a variable in the "Variable to receive SR Handle" field).

## ACTION STEPS

None.

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

Dialogic  
NMS Communications

## RELATED STEPS



[SR Get Digits](#)



[SR Get String](#)



[SR Get Yes/No](#)

### 5.6.1.10.8 SR Get Digits

Collects a series of digits using speech recognition.

## PROPERTIES

**Variable to receive voice digits:** The result of the speech recognition - a series of digits - will be put in this TeleFlow variable. This field is populated with @SR\_DIGITS by default, but may be any TeleFlow variable. This field must not be blank.

**THRESHOLD confidence %:** The result confidence is compared against this minimum confidence to determine a valid result. This field is populated with 50 by default. This field must not be blank. (Confidence is an indication of how "certain" the SR Engine is that the result found is a match to what the caller said.)

**Variable to receive confidence %:** The result confidence percentage will be put in this TeleFlow variable. This field is populated with @SR\_CONFIDENCE by default. This field can be blank.

**No-Voice TIMEOUT (ms):** If no speech or touch-tones are detected in this time, the timeout path is followed. Note that if speech or touch-tones are pending processing when the step is reached, this timeout will not occur. This field is blank by default. If this field is blank, the system default @TIMEOUT setting will be used.


**Remember this recognition for verification through SpeechTrainer:** If checked, the


recognition is made available for review with SpeechTrainer. ('Save SR Audio to Log Directory' must be checked in [TeleFlow Config](#) for the appropriate Speech Trainer files to be generated.)

**Variable to receive SR Handle:** A variable to receive an SR Handle generated by TeleFlow. In order to flag an SR result (using [SR Flag Result](#)), an SR Handle is required.

**Retain audio on THRESHOLD:** Check this checkbox to compare the speech recognition audio against another grammar (such as SR Get String) when the result confidence is less than the THRESHOLD confidence.

## ACTION STEPS

 [Unrecognized Voice Action](#): The confidence was less than the THRESHOLD confidence, or a touch tone was received, and the confidence was set to 0.

 [Time Out](#): No speech or touch tones were detected before the No-Voice TIMEOUT elapsed.

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

Dialogic (prompted only)  
NMS Communications

## RELATED STEPS



[SR Acquire](#)



[SR Get Yes/No](#)



[SR Get String](#)

### 5.6.1.10.9 SR Get String

Collects a response through speech recognition. Valid responses may be words or short phrases, matched against a grammar.

## PROPERTIES

**Variable to receive response:** The result of the speech recognition will be put in this TeleFlow variable. This field is populated with @SR\_STRING by default, but may be any TeleFlow variable. This field must not be blank.

**THRESHOLD confidence %:** The result confidence is compared against this minimum confidence to determine a valid result. This field is populated with 50 by default. This field must not be blank.

**Variable to receive confidence %:** The result confidence will be put in this TeleFlow variable. This field is populated with @SR\_CONFIDENCE by default. This field can be blank.

**No-Voice TIMEOUT (ms):** If no speech or touch-tones are detected in this time, the timeout path is followed. Note that if speech or touch-tones are pending processing when the step is reached,

this timeout will not occur. This field is blank by default. If this field is blank, the system default @TIMEOUT setting will be used.

**Remember this recognition for verification through SpeechTrainer:** If checked, the recognition is made available for review with SpeechTrainer. ('Save SR Audio to Log Directory' must be checked in [TeleFlow Config](#) for the appropriate Speech Trainer files to be generated.)


**Variable to receive SR Handle:** A variable to receive an SR Handle generated by TeleFlow. In order to flag an SR result (using [SR Flag Result](#)), an SR Handle is required.


**Retain audio on THRESHOLD:** Check this checkbox to compare the speech recognition audio against another grammar (such as SR Get Yes/No or SR Get Digits) when the result confidence is less than the THRESHOLD confidence.

**Base grammar (name or file):** This field specifies the name of the [Nuance](#) pre-compiled grammar, or the SRGS grammar file for use with [LumenVox](#). This field must not be blank for Nuance, but may be blank for LumenVox. If this field is left blank, only the additional grammar text will be used for speech recognition. If both this field and the additional grammar field are left blank, the application will terminate with a FATAL error.

**Additional grammar text:** This field specifies SRGS grammar rules in addition to the base grammar, but can only be used by [LumenVox](#). If this field is left blank, only the base grammar will be used for speech recognition. If both this field and the additional grammar field are left blank, the application will terminate with a FATAL error.

## ACTION STEPS

 [Unrecognized Voice Action](#): The confidence was less than the THRESHOLD confidence, or a touch tone was received, and the confidence was set to 0.

 [Time Out](#): No speech or touch tones were detected before the No-Voice TIMEOUT elapsed.

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

Dialogic (prompted only)  
NMS Communications

## RELATED STEPS



[SR Acquire](#)



[SR Get Digits](#)



[SR Get Yes/No](#)

## 5.6.1.10.10 SR Get Yes/No

Collects a "yes" or "no" response through speech recognition

**PROPERTIES**

**Variable to receive response:** The result of the speech recognition, "yes" or "no", will be put in this TeleFlow variable. This field is populated with @SR\_YN by default, but may be any TeleFlow variable. This field can be blank.

**THRESHOLD confidence %:** The result confidence is compared against this minimum confidence to determine a valid result. This field is populated with 50 by default. This field must not be blank.

**Variable to receive confidence %:** The result confidence will be put in this TeleFlow variable. This field is populated with @SR\_CONFIDENCE by default. This field can be blank.

**No-Voice TIMEOUT (ms):** If no speech or touch-tones are detected in this time, the timeout path is followed. Note that if speech or touch-tones are pending processing when the step is reached, this timeout will not occur. This field is blank by default. If this field is blank, the system default @TIMEOUT setting will be used.

**Remember this recognition for verification through SpeechTrainer:** If checked, the recognition is made available for review with SpeechTrainer. ('Save SR Audio to Log Directory' must be checked in [TeleFlow Config](#) for the appropriate Speech Trainer files to be generated.)

**Variable to receive SR Handle:** A variable to receive an SR Handle generated by TeleFlow. In order to flag an SR result (using [SR Flag Result](#)), an SR Handle is required.

**Retain audio on THRESHOLD:** Check this checkbox to compare the speech recognition audio against another grammar (such as SR Get String or SR Get Digits) when the result confidence is less than the THRESHOLD confidence.

**ACTION STEPS**

[Yes](#): "yes" was detected with adequate confidence.



[No](#): "no" was detected with adequate confidence.



[Unrecognized Voice Action](#): The confidence was less than the THRESHOLD confidence, or a touch tone was received, and the confidence was set to 0.



[Time Out](#): No speech or touch tones were detected before the No-Voice TIMEOUT elapsed.

**SUPPORTED TELEPHONY HARDWARE PLATFORMS**

Dialogic (prompted only)  
NMS Communications

**RELATED STEPS**



[SR Acquire](#)



[SR Get Digits](#)



[SR Get String](#)

#### 5.6.1.10.11 SR Release

Releases resources acquired by SR Acquire step.

This step has no effect if no resources are currently retained.

#### PROPERTIES

None.

#### ACTION STEPS

None.

#### SUPPORTED TELEPHONY HARDWARE PLATFORMS

Dialogic  
NMS Communications

#### RELATED STEPS



[SR Acquire](#)



[SR Barge-In On](#)



[SR Barge-In Off](#)

#### 5.6.1.10.12 SR VoicePass Auth

Authenticates against a biometric voice password.

With this step, you can challenge a caller to speak his or her unique voice password, and authenticate the utterance against a voice-print pattern previously created with the [SR VoicePass Reg](#) step. This process is analogous to entering a password (or PIN) with a keypad, and comparing against a stored password.

Note: Voice Password authentication is not available as a part of the standard TeleFlow Suite. You must purchase additional software to use this step.

#### PROPERTIES

**Voice pattern filename or User ID:** File name where the previously registered voice-print pattern

is stored. Typically, a user's unique identifier is used as the file name to make it easier to match an ID to its associated voice-print pattern. The pattern specified will be used to match against the utterance recorded from the current caller to determine if there is a match.

**Do not play start recording tone (beep):** If checked, this indicates the step should not play a tone/beep just prior to recording the caller's password.

**Authentication Factor:** The authentication factor controls the level of consistency required to match between the caller's utterance and the voice-print pattern. The lowest setting yields the strictest control (even small inconsistencies would be detected), while the highest setting yields more lenient control (some inconsistencies and noise are ignored).

The mid-range setting (50%) is recommended for most scenarios, providing good matching accuracy while filtering out some background noise and natural inconsistencies in a person's speech.

**Maximum time recordable:** Sets the maximum recording time in milliseconds for the voice password utterance. The default is 4000 (4 seconds).

## ACTION STEPS



**True:** The caller's voice and password match the biometric voice-print pattern. The caller has been authenticated.



**False:** Either the caller's voice or the password spoken do not match the biometric voice-print pattern. The caller could not be authenticated, and should be considered unauthorized.



**Fail:** The authentication process failed. The most likely cause is an invalid path or file name in the "Voice pattern filename" field.



**Timeout:** Path followed if caller doesn't speak during the utterance recording.

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

Ai-Logix  
Dialogic  
NMS Communications

## RELATED STEPS



[SR VoicePass Reg](#)

5.6.1.10.13 SR VoicePass Reg

Registers a biometric voice password.

This step allows a user to create his or her unique voice-print that can be used later to authenticate the user. This process involves taking 3 samples of the user's voice, speaking the exact same password (or phrase) each time. This process is analogous to creating a password or PIN for the first time. The results are compiled into a "pattern" and stored in a file that is later used to authenticate log-in attempts.

Note: Voice Password authentication is not available as a part of the standard TeleFlow Suite. You must purchase additional software to use this step.



## PROPERTIES

**Voice pattern filename (or User ID):** File name to store the voice-print pattern created by this step. Typically, the user's ID is also used as the file name to make it easier to match an ID to its associated voice-print pattern.

**Do not play start recording tone (beep):** If checked, this indicates the SR VoicePass Reg step should not play a tone/beep just prior to recording for each voice-print.

**Play voice file / code for recording 1 of 3:** The Script Code, or name of an audio file, to play to prompt the caller for the first utterance of his or her voice password.

**Play voice file / code for recording 2 of 3:** The Script Code, or name of an audio file, to play to prompt the caller for the second utterance of his or her voice password.

**Play voice file / code for recording 3 of 3:** The Script Code, or name of an audio file, to play to prompt the caller for the third utterance of his or her voice password.

**Registration Factor (1-tight - 100-loose):** The registration factor controls consistency checking between the password utterances as they are processed and compiled to form a voice print pattern. The lowest setting yields the strictest control; even small inconsistencies would be detected. The highest setting produces the most lenient control; some inconsistencies and noise are ignored.

The mid-range setting (50%) is recommended for most scenarios, providing good matching accuracy while filtering out some background noise and natural inconsistencies in a person's speech.

**Maximum time recordable (ms):** Defines the maximum recording length in milliseconds of each voice password utterance. The default is 4000 (4 seconds).

## ACTION STEPS



**Time Out:** Path followed if caller doesn't speak during the utterance recording.



**Fail:** Registration failed. Likely causes: The caller didn't speak consistently enough to successfully register, or one of the "Play voice file" entries was invalid.

## SUPPORTED TELEPHONY HARDWARE PLATFORMS

Ai-Logix  
Dialogic  
NMS Communications

## RELATED STEPS



[SR VoicePass Auth](#)

### 5.6.1.11 Web Steps

The following list is comprised of all of the Web/network related steps that are available in TeleFlow Designer. These steps are found under the Web tab of the [Steps Toolbox Window](#).

[HTTP](#)  
[Send Email](#)  
[TCP Connect](#)  
[TCP Disconnect](#)  
[TCP Receive](#)  
[TCP Send](#)  
[UDP Bind](#)  
[UDP Close](#)  
[UDP Receive](#)  
[UDP Send](#)

#### 5.6.1.11.1 HTTP

This step opens a TCP/IP connection to an HTTP server, issues an HTTP 1.0 command, receives the response from the HTTP server, and then closes the TCP/IP connection, during all of which an audio file may optionally be played. This step supports 7 different HTTP commands: GET, POST, PUT, DELETE, OPTIONS, HEAD, and TRACE.

**HTTPS:** HTTPS connections are assumed when the URL contains "https".

#### PROPERTIES

**URL:** The URL of HTTP command. The http:// prefix is not required but may be used. The host name or IP address may include a port specifier, otherwise the default HTTP port, 80, is used.

**HTTP Request Type:** Sets the HTTP request type to use. The default is GET, to load an HTTP document. Understanding what the various types of HTTP requests are for requires a solid understanding of HTTP.

**HTTP Request Message Body:** Body text (to send) for HTTP commands that require additional information not expressed in the URL.

**Variable to load HTTP Response Header into:** TeleFlow variable to load with the header of the HTTP command response, or any socket error messages.

**Variable to load HTTP Response Body into:** TeleFlow variable to load with the body of the HTTP command response. In the case of an HTTP GET, this would be the document retrieved from the HTTP server. (Simply put: This could be the html contents of a web page)


**HTTP Initial Connection TIMEOUT:** When connecting to an HTTP server, this is longest time to wait, in ms, before giving up/timing out on the connection attempt. This field defaults to 500 ms. A FATAL error is generated if this field evaluates to less than 1 ms.


**HTTP Initial Connection Retries:** This is the number of times a connection will be retried before following the TIMEOUT path. If this field is left blank, then a default of 2 retries is used.

**Silence Timeout (ms):** An overall timeout for the step. This is the maximum time to wait, in ms, before following the TIMEOUT path. This is called a "Silence Timeout", because while the step is retrieving a result, the caller is likely not hearing any audio.

**Play Voice Script (optional):** Optional audio to play while a caller is waiting for the HTTP command response. This feature is not available on the Dialogic platform.

## ACTION STEPS

 **Fail:** The FAIL path is followed when the URL HTTP server cannot be resolved or there is a socket error. Any response that has been received so far will be loaded into the HTTP Response Body variable.

 **Time Out:** The TIMEOUT path is followed when the HTTP server does not connect within the connection retries and timeout specified, or the HTTP server does not send the complete HTTP command response within the silence timeout specified. Any response that has been received so far will be loaded into the HTTP Response Body variable.

## RELATED STEPS



[Load Text](#)



[Send E-mail](#)



[TCP Connect](#)

### 5.6.1.11.2 Send Email

Sends a plaintext email message.

**NOTE:** Since the @ symbol is used to indicate a TeleFlow variable, any literal entry that should contain an @ must be entered with double @'s. (eg: name@@company.com)

## PROPERTIES

**Mail Server:** The SMTP server to send mail through.

**SMTP authentication method:** Choose the appropriate authentication method radio button for the SMTP server. The default - 'None' - requires no entries in the Sender's Account field or the Sender's Password field.

**Sender's Address:** A valid account that is known to the SMTP server and is allowed to send mail. (eg: @SENDERS\_ADDRESS or name@@company.com)

**Sender's password:** The SMTP server account password. A FATAL error will be generated if this field is blank and either the 'LOGIN' or 'PLAIN' SMTP authentication radio buttons are selected.

**From address:** The address the email is coming from, not necessarily the account that the message is being sent with. (e.g. @SENDERS\_ADDRESS or name@@company.com). This is useful when sending via a particular account (eg. ivr@company.com) but replies should go to another account (eg. sales@company.com). Quotes and angle brackets may be used as per RFC 2822. A FATAL error will be generated if this field evaluates to empty and either the 'LOGIN' or 'PLAIN' SMTP authentication radio buttons are selected.

**Recipients:** A list of one or more recipients that the email is to go to. If more than one recipient is required, they may be separated by commas or semi-colons.

**Subject line:** Email subject line.

**Body Text:** Body text of the message.

**Attachments (separate file names with commas):** A comma or semi-colon separated list of files to be attached. File paths can be either relative to the TeleFlow application or absolute.

## ACTION STEPS



**Fail:** The email send failed. See the application log for failure details.

### 5.6.1.11.3 TCP Connect

Establishes a TCP connection with a TCP/IP host (or peer).

## PROPERTIES

**Host Name or IP Address:** Host name or IP address of the TCP/IP host to establish a connection with. The host name or IP address may optionally be followed by a port specifier e.g. www.enGenic.com:80. If the Host Port # field is not left empty, the Host Port# will override any port specified in this field.

**Host port #:** Port of TCP/IP host to establish a connection with. This field may be left blank if the port is specified in the Host Name or IP Address field. This field will override any port specified in the Host Name or IP Address field.

**Use SSL?:** Check this box if you are connecting to a TCP server using secure sockets layer.

**TIMEOUT (ms):** Time to wait for a connection, in milliseconds, before following the TIMEOUT path. If left blank, a value of 500 ms is used.

**Variable to receive Socket Handle:** TeleFlow variable to accept the socket handle generated by the TCP connection. This variable stores the socket handle for use with other TCP steps. A FATAL error will be generated if this field is left blank.

## ACTION STEPS



**Time Out:** Unable to connect in the time specified.



**Fail:** Unable to resolve host name or IP address, or to acquire a socket from the operating system. Please see the log for failure details.

## RELATED STEPS



[TCP Disconnect](#)



[TCP Receive](#)



[TCP Send](#)

#### 5.6.1.11.4 TCP Disconnect

Disconnect from a TCP/IP host (or peer).

#### PROPERTIES

**Socket Handle:** Socket handle(generated by the [TCP Connect](#) step) to disconnect.

#### ACTION STEPS

None.

#### RELATED STEPS



[TCP Connect](#)



[TCP Receive](#)



[TCP Send](#)

#### 5.6.1.11.5 TCP Receive

Receive data from a TCP/IP host (or peer) across a TCP connection

#### PROPERTIES

**Socket Handle:** Socket handle(generated by the [TCP Connect](#) step) that specifies the connection to use to receive data.

**Variable to receive Message:** TeleFlow variable to load with data received across TCP connection. A FATAL error will be generated if this field is blank.

**TIMEOUT (ms):** Delay before following the TIMEOUT path when no exit condition occurs, such as minimum/maximum length of characters received, or termination string received.

If left blank, the value of @TIMEOUT (default 3000) is used.

**Minimum length message to receive:** TCP/IP Receive checks after each TCP/IP data packet to see if at least the minimum number of characters has been received, and proceeds to the next step if they have.

If left blank, TCP/IP Receive will continue receiving data until the maximum length of characters is received, the termination string is received, or a timeout occurs.

Length is specified in bytes.

**Maximum length message to receive:** TCP/IP Receive will proceed to the next step if the

maximum number of characters has been received.

If left blank, TCP/IP Receive will continue receiving data until the minimum length of characters is received, the termination string is received, or a timeout occurs.

Length is specified in bytes.

**Termination string (define end of record):** TCP/IP Receive will stop receiving data and proceed to the next step if the termination string is detected. If left blank, TCP/IP Receive will continue receiving data until the minimum or the maximum length of characters is received, or a timeout occurs

**Include termination string in Message:** Checking this checkbox indicates that the data loaded into the 'Variable to receive Message' should include the termination string when it occurs.

**Hexadecimal representation:** Checking this checkbox indicates that the data loaded into the 'Variable to receive Message' should be represented as a sequence of hexadecimal nibbles. Use this setting to receive binary data across your TCP connection.

Note that 2 nibbles represent 1 byte of data.

**Upper case:** Checking this checkbox indicates that the hexadecimal nibbles should use UPPER case('A'..'Z').

## NOTES

If a NULL character is received and hexadecimal representation is not being used, TCP/IP Receive will print a warning in the log and proceed to the next step but no error will be generated; no data other than the null character will be lost.

## ACTION STEPS



**Time Out:** No message received in time specified.



**Fail:** Unable to receive message, for a cause other than timeout, usually due to an invalid socket or a socket error.

## RELATED STEPS



[TCP Connect](#)



[TCP Disconnect](#)



[TCP Send](#)

### 5.6.1.11.6 TCP Send

Send data to a TCP/IP host (or peer) across a TCP connection


## PROPERTIES

**Socket Handle:** Socket handle(generated by the [TCP Connect](#) step) that specifies the connection to send data across.

**Message to send:** Data to send to across the TCP connection.

**Hexadecimal representation:** Checking this checkbox indicates that the 'Message to send' field data should be interpreted as a sequence of hexadecimal nibbles and converted appropriately for transmission. 2 nibbles represent 1 byte of data. The nibbles are case insensitive.

## ACTION STEPS

 [Fail](#): Unable to send message, usually due to an invalid socket handle or a socket error.

## RELATED STEPS



[TCP Connect](#)



[TCP Disconnect](#)



[TCP Receive](#)

### 5.6.1.11.7 UDP Bind


Binds (reserves) a local port for use by the TeleFlow application, and return a socket handle that can be used in your application to receive or send data via UDP.

## PROPERTIES

**Host port #:** Local port number to bind to.

**Variable to receive Socket Handle:** TeleFlow variable to accept the handle generated by the UDP port binding. This stores the socket handle for use with other UDP steps. A FATAL error will be generated if this field is left blank.

## ACTION STEPS

 [Fail](#): Unable to bind to the port, or to acquire a socket from the operating system. Please see the log for failure details.

## RELATED STEPS



[UDP Close](#)



[UDP Receive](#)



### [UDP Send](#)

#### 5.6.1.11.8 UDP Close

Closes/releases a previously bound local port.

#### PROPERTIES

**Socket Handle:** Socket handle (generated by the [UDP Bind](#) step) to be disconnected.

#### ACTION STEPS

None.

#### RELATED STEPS



### [UDP Bind](#)



### [UDP Receive](#)



### [UDP Send](#)

#### 5.6.1.11.9 UDP Receive

Listens for incoming packets on a particular port (specified by the socket handle of a previous [UDP Bind](#) step), and receives a packet of data.

#### PROPERTIES

**Socket Handle:** Socket handle (generated by the [UDP Bind](#) step) to listen to.

**Variable to receive Peer:** TeleFlow variable to receive the host name or IP address of the sending peer.

**Variable to receive Port:** TeleFlow variable to store the remote peer's sending port, as received in the UDP packet.

**Variable to receive Message:** TeleFlow variable to receive the message data into.

**TIMEOUT (ms):** Time (in milliseconds) to wait for a message to be received before following the TIMEOUT path. If left blank, the current value of @TIMEOUT (default 3000) is used.

**Hexadecimal representation:** Checking this checkbox indicates that the data loaded into the 'Variable to receive Message' should be represented as a sequence of hexadecimal nibbles. Use this setting to receive binary data. Note that 2 nibbles represent 1 byte of data.

**Upper case:** Checking this checkbox indicates that the hexadecimal nibbles should use UPPER case ('A'..'F').



## ACTION STEPS



**Time Out:** No message received in the time specified.



**Fail:** Unable to receive message, for a cause other than timeout, typically due to an invalid socket or a socket error. Please see the log for failure details.

## RELATED STEPS



[UDP Bind](#)



[UDP Close](#)



[UDP Send](#)

### 5.6.1.11.10 UDP Send

Sends data via UDP, using a port previously bound with the [UDP Bind](#) step.

## PROPERTIES

**Socket Handle:** Socket handle (generated by the UDP Bind step) to use to send.

**Peer host name or IP address:** Host name or IP address of the peer to send the message to.

**Peer Port #:** The peer's (destination) port number to send the message to.

**Message to send:** Message or data payload to send.

**Hexadecimal representation:** Checking this checkbox indicates that the 'Message to send' field data should be interpreted as a sequence of hexadecimal nibbles and converted appropriately for transmission. Use this setting to send binary data. 2 nibbles represent 1 byte of data. The nibbles are not case-sensitive.

## ACTION STEPS



**Fail:** Unable to send the message due to an invalid socket or a socket error. Please see the log for failure details.

## RELATED STEPS



[UDP Bind](#)



[UDP Close](#)



[UDP Receive](#)

5.6.1.11.11 WebStepsTOC

[HTTP](#)  
[Send Email](#)  
[TCP Connect](#)  
[TCP Disconnect](#)  
[TCP Receive](#)  
[TCP Send](#)  
[UDP Bind](#)  
[UDP Close](#)  
[UDP Receive](#)  
[UDP Send](#)

## 5.7 Variables

This chapter covers how TeleFlow variables (named "containers" for storing data in a TeleFlow application) work. The following chapters cover TeleFlow variables in detail.

[Variable Basics and Scope](#)  
[Variable Data Types](#)  
[Arrays](#)  
[System Variables](#)  
[Application Parameters @P1 ... @P9](#)

### 5.7.1 Variable Basics and Scope

In programming, a variable is a named container for storing data. To change, manipulate, or read/check the data, a program refers to the container by its name.

The following chapter sub-topics cover the things a TeleFlow application developer should know about variables in TeleFlow. Although they are referred to as "the basics", some of the information below can be a little overwhelming. One recommendation for dealing with this is to read the information carefully once, and re-read it from time to time until you have a solid understanding of all the concepts presented here.

#### TELEFLOW VARIABLES - THE BASICS

**Variable Names:** TeleFlow variable names start with an '@' symbol, and are not case sensitive (in fact, they are all treated as fully UPPER CASE by TeleFlow). Variable names can contain any alpha-numeric characters, and the underscore "\_". A user-defined variable is otherwise not restricted in terms of what it can be named, except that [System Variables](#), and [Application Parameters Variables](#) are all reserved. Although not strictly necessary, it is a good idea to name all variables in ALL UPPER CASE for application readability: Since the automatic documentation for steps in TeleFlow Designer is in mixed case, variables stand out better in upper case.

Valid variable name examples: @MY\_VARIABLE, @my\_variable (valid, but discouraged), @MY\_OTHER\_VARIABLE

Invalid variable name examples: my\_variable (no preceding @ sign), @&&my\_variable (contains non-alpha-numeric special characters), @my variable (contains a space)

**Variable Creation:** A TeleFlow variable can be created by ANY step that sets a variable to a value. This "instant variable creation" means that a TeleFlow variable doesn't have to be specifically created (such as by a [Set Global Variable](#) or [Set Local Variable](#) step) before it is used. If, for example, you wish to create a variable name @TT\_INPUT and use it in a [Get TT String](#) step to capture a caller's touch-tone input, you can simply enter @TT\_INPUT into the "Variable to receive touch tone(s)" field. TeleFlow will create and populate the variable (as a [Global Variable](#))

with the touch tone input of the caller.

Similarly, SQL Statement and SQL Fetch will perform "automatic variable creation" when retrieving rows from a query. The fields requested in the SELECT portion of the query will result in global variables of the same name as the field or alias (See [SQL Fetch](#) for more information on how this works).

Note that variables are created when the step that creates them is reached in the call or application flow. TeleFlow variables do not have strict data-types, but does treat variables as certain types based on context (i.e. TeleFlow decides what data type a variable is based on what the application attempts to do with it). The [Variable Data Types](#) chapter describes this in greater detail.

**Flowchart Parameters:** Any TeleFlow flowchart can receive parameter variables from the flowchart that runs it. All variables received in such a way are automatically [local variables](#) in the flowchart that receives them. The receiving variables and the way in which they are received are defined in a flowchart's [START step](#). The values to pass into the flowchart are set in the [Run Flowchart](#) step. A flowchart parameter variable can be "Passed by value" or "Passed by reference".

Pass by value: The receiving variable for any parameter passed by value is simply set to the value passed in. If nothing is supplied in the Run Flowchart step, the receiving variable is set to empty string.

Pass by reference: When a flowchart reaches the [FINISH step](#), any variable passed by reference is updated in the flowchart that passed it in. Eg. @MY\_VARIABLE contains 5, and is passed by reference into flowchart 'Add\_One' in parameter 1, which receives the variable as @NUMBER\_TO\_INCREMENT. The 'Add\_One' flowchart changes @NUMBER\_TO\_INCREMENT from 1 to 2, then the flowchart exits. Because the variable was passed by reference, @MY\_VARIABLE is updated to 2.

**Application Parameters:** An application can be configured or run such that it receives application parameters in variables named @P1 through @P9. See the [Application Parameters @P1 ... @P9](#) topic for details.

**System Variables:** System Variables are variables created by TeleFlow. Most are read-only variables that cannot be set by an application, but provide an easy way to get otherwise unavailable information into an application. The others are settings variables that allow the application to change how TeleFlow performs certain tasks. The help chapter on [System Variables](#) has the complete list of system variables and how they can be used.

## VARIABLE SCOPE - LOCAL AND GLOBAL VARIABLES

As mentioned in THE BASICS, a TeleFlow variable doesn't exist until the application flow has passed through the step that creates it. Another concept important to understand in TeleFlow is that of variable scope.

**Local Variables:** Local variables exist only in the flowchart in which they are created. If a flowchart#1 uses [Run Flowchart](#) to run flowchart#2, flowchart#2 doesn't have access to variables created in flowchart#1. However, when flowchart#2 reaches its FINISH step and returns to flowchart#1, flowchart#1 can continue to use local variables it previously created. If a flowchart#2 needs access to a local variable created in flowchart#1, flowchart parameters (see above) can be used to pass variables into a flowchart.

If a local variable has the same name as a global variable (this is generally a very bad idea, as it can lead to a great deal of confusion), the local variable takes precedence over the global. This

means that in the flowchart the local variable exists, any time an operation is done with a variable of the duplicate variable name, it is assumed to be the local variable (Unless the global is specifically set with a Set Global Variable step).

**Global Variables:** Global variables are available anywhere in an application from the point at which they are created. Any variable that is not specifically created as a local variable - with a [Set Local Variable](#) step, the "LOCAL" tag on a [Run Basic Script](#) step, or as a flowchart parameter variable (see above) - is created as a global variable.

**Global Variables and using Run Tap:** If a TeleFlow application runs another TeleFlow application using the [Run TAP](#) step, global variables created in the "parent application" (I.e. the TAP that ran another TAP) don't exist / aren't available in the application it runs. If it is necessary for the "child application" to access a variable created in the parent, the parent must pass the variable(s) to the child using the "Parameter" fields of the Run TAP step. The child application will then receive the global variables as parameter variables [@P1 through @P9](#), which it can then assign to its own global variables, or use directly for whatever purpose it needs to.

### TIP - SETTING APPLICATION GLOBALS IN ONE PLACE

A typical IVR application may require the use of several different user defined global variables. In order to manage variables efficiently, try to set aside a specific flowchart that does nothing but create and set defaults for variables that you wish to use throughout your application. A variable does not need to be used immediately after you place the [Set Global Variable](#) step. If you 'create' all of your variables in one flowchart, you can use them as you need them throughout other flowcharts. By using this method, you will always be able to find the original Set Global Variable steps whenever you need to change the default. The flowchart in which you set these defaults is also an ideal place to document - using the comments section of the step properties - what each global variable is for.

If you look at most of the example applications that are included in the TeleFlow download, a flowchart named 'Defaults' (the flowchart name is arbitrary, and could just as easily be named 'Variables,' etc) is used to set up most or all of the application's global variables. The flowchart is typically called early in the application, so that each time the application or call restarts, all of the default values for the variables are set and waiting for use. To ensure that the variables are reset for each new caller, you may want to [call](#) the default variables flowchart immediately after a [Wait for Call](#) or [Place Call](#) step.

## 5.7.2 Variable Data Types

TeleFlow does not have strict variable data types. In fact, every variable in TeleFlow is basically treated as a string. How the variable is used, manipulated, or output (I.e. Using a [Say](#) step to play data to a caller, such as the date) determines what data type it is. Note that nothing "sets" a TeleFlow variable to a particular type, it is simply treated like it is a particular type for purposes of the task being performed (I.e. If you have two variables, one containing "1" and the other containing "1", and you add them together using the [Math](#) step, they will be treated as integers and added together, resulting in "2". If you instead concatenate them together, they are treated as strings, and the result would be "11")

Since there are no specific data types, it is most important understand what data types a variable might be "assumed" to be by TeleFlow steps, and how to format variables to ensure they are treated how an application needs them to. The following sections describe the "data types" TeleFlow can assume a variable to be, and how you can take advantage of them.

The data types TeleFlow will assume are:

- [String variables](#)
- [Date variables](#)
- Date/Time variables

- [Numeric variables](#)
- [Time variables](#)

## STRING VARIABLES

**Format:** No special formatting is required. All TeleFlow variables are strings by default, and are only treated as anything else if the context of the operation demands it. For example, using a [Compare](#) step to compare which of two variables containing values "1" and "20" is greater will compare the two value numerically, and "20" will evaluate as the greater value.

**Description/Uses:** String variables are variables that contain any characters from the alphanumeric character set. There is no upper limit to the number of characters a string variable can contain. The following is a short list of operations you can perform on strings in TeleFlow:

- Get the string length of a string variable using the "String length of..." option of the [Math](#) step.
- Copy a portion of a string variables using the "Copy string from..." option of the [Math](#) step.
- Concatenate multiple string variables together. A TeleFlow developer can take advantage of a powerful feature of TeleFlow to do this: TeleFlow replaces all variables in all fields - except any field where the variable is being set to a value based on the results of the step, of course - with their values before performing any other the operation. This means that in any step, an application can concatenate a series of variables, mixed in with literal text, and TeleFlow will put it all together. Here is a simple example of how this could work:
  - @MY\_VAR contains "example"
  - @MY\_OTHER\_VAR contains "substitution"
  - If the contents of the "Message test" field of a [Write Text](#) text step in an application are "This is my @MY\_VAR of how TeleFlow variable @MY\_OTHER\_VAR works.", TeleFlow will display this information in the log as "This is my example of how TeleFlow variable substitution works."

**Example:** A STRING OF CHARACTERS

## DATE VARIABLES

**Format:** YYYY/MM/DD. To have TeleFlow treat a variable as a date, it must adhere to the YYYY/MM/DD format (four-digit year, two-digit month, two-digit day, which each section separated by forward-slashes).

**Description/Uses:** Having a TeleFlow variable in date format allows you to do some of the following:

- Speak a date to a caller using the [Say](#) step.
- Perform date math by adding or subtracting whole numbers to or from a date variable. TeleFlow date math uses a precision of 1 day, so all numbers to deduct or add must be expressed in days. This will change the date to a later or earlier date, respectively. This can be done with the [Math](#) step.
- Use the [Compare](#) step to see which of two dates is greater.
- NOTE: The TeleFlow system variable [@DATE](#) provides the current date in the TeleFlow date format.

**Example:** 2007/09/14

## DATE/TIME VARIABLES

**Format:** YYYY/MM/DD HH:MM:SS. The TeleFlow date/time format is the same a [TeleFlow Date](#)

followed by a single space and a [TeleFlow Time](#). The [@DATE\\_SEPARATOR](#) system variable can be set to change the date separator to "/" (the default) or "-" (recommended for compatibility with ISO 8601)

**Description/Uses:** Representing a single point in time by a date/time variable makes some calculations easier.

**Example:** 2009/03/10 16:18:34

## NUMERIC VARIABLES

**Format:** Data in integer or real variables must be numeric only, including decimal points.

**Description/Uses:** Numeric variables can contain any valid number (Range: -2147483648 to 2147483647). For simplicity, when [Math](#) is performed on a numeric variable, the number with the greatest number of decimal places determines the number of decimal places for the result.

**Example:** 10

## TIME VARIABLES

**Format:** HH:MM:SS. To have TeleFlow treat a variable as a time, it must adhere to the HH:MM:SS format (two-digit hour using the 24-hour clock, two-digit minute, two-digit seconds, with each section separated by colons).

**Description:** Having a TeleFlow variable in time format allows you to do some of the following:

- Speak a time to a caller using the [Say](#) step.
- Perform time math by adding or subtracting whole numbers to or from a time variable. TeleFlow time math uses a precision of 1 second, so all numbers to deduct or add must be expressed in seconds. This will change the time to a later or earlier time, respectively. Time math "wraps" around midnight, in either direction. So, if the time you are working with is 23:59:59 and you add two seconds, the number would be 00:00:01. This can be done with the [Math](#) step.
- Use the [Compare](#) step to see which of two times is greater.
- NOTE: The TeleFlow system variable [@TIME](#) provides the current time in the TeleFlow time format.

**Example:** 09:42:12

### 5.7.3 Arrays

TeleFlow doesn't have traditional arrays, but an application developer can create logical arrays by putting array elements into a delimited string (i.e. You can populate a variable with elements of an array, separating each element with a special character, such as a comma, to create a "logical array"). You can then use the [Parse](#) step to search the delimited string for a desired - numbered - element.

To use the Parse step to return an element of a delimited string, follow these steps:

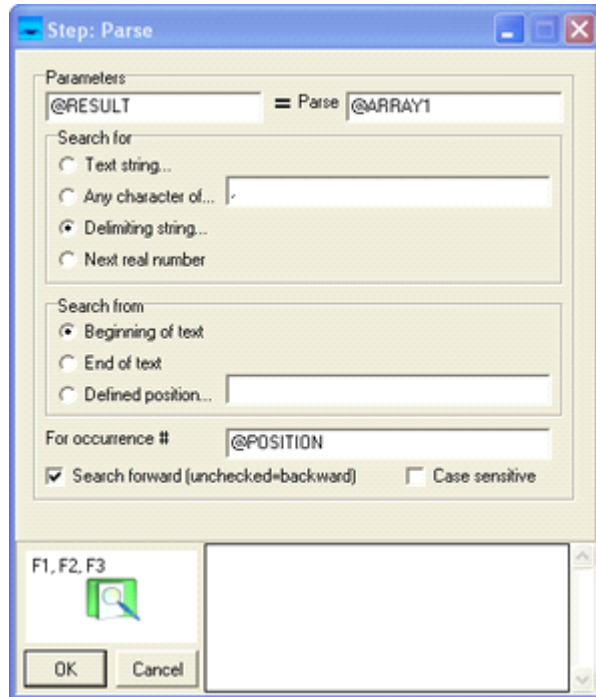
1. Set a result variable to receive the array element.
2. Populate the "Parse" field with the variable containing the logical array.
3. Set the "Search for" radio button to "Delimited String"
4. Enter the delimiter used in the string to divide up the elements into the search string field in the "Search for" panel.
5. Set the "Search from" radio button to "Beginning of text".
6. Enter the element number desired in the "For occurrence #" field.

To read backwards from the end of a delimited string, uncheck the "Search forward" checkbox, and choose the "End of text" option in the "Search from" panel.

### Example:

In the example shown below, assume the following:

- @ARRAY1 = apples,oranges,bananas,peaches
- @POSITION = 2



Given the example shown, @RESULT would be 'oranges', because 'oranges' is the second element in the comma-delimited "array".

## 5.7.4 System Variables

System Variables are variables created by TeleFlow. Most are read-only variables that cannot be set by an application, but provide an easy way to get otherwise unavailable information into an application. The others are settings variables that allow the application to change how TeleFlow performs certain tasks.

The system variable topics use the following layout:

**Description:** Describes what the variable is/what it is for.

**Date Type:** The TeleFlow data type of the variable. See [Variable Data Types](#) for information on how data types are handled in TeleFlow.

**Sample Uses:** OPTIONAL. Some system variable topics provide examples of how the system variable might be used in a TeleFlow application.

### READ-ONLY SYSTEM VARIABLES

The following variables cannot be set by a TeleFlow Application (Although @P1 through @P9 are set prior to an application running), but can be used to provide information to an application.

[Application Parameters @P1 ... @P9](#)  
[@CALL\\_ELAPSE](#)  
[@CALL\\_END](#)  
[@CALL\\_START](#)  
[@CALLOUT\\_REASON](#)  
[@CALLOUT\\_RESULT](#)  
[@CHAR or Character variables](#)  
[@DATE](#)  
[@DATE.UTC](#)  
[@DATETIME](#)  
[@DATETIME.UTC](#)  
[@FORMFEED](#)  
[@HOOK\\_STATUS](#)  
[@LINEFEED](#)  
[@P1 to @P9](#)  
[@PORT](#)  
[@RECORD\\_RESULT](#)  
[@SPACE](#)  
[@SPELLTONES](#)  
[@SYS\\_APP\\_STATUS](#)  
[@SYS\\_CALLREF](#)  
[@SYSDIR](#)  
[@SYSDRIVE](#)  
[@TIME](#)  
[@TIME.UTC](#)  
[@TRUNK\\_TYPE](#)  
[@UNIX\\_TIME](#)  
[@WEEKDAY](#)

#### SETTINGS SYSTEM VARIABLES

The following variables affect how a TeleFlow application behaves under certain circumstances. Some of these cannot be set directly, but rather, must be set through specific TeleFlow steps. This will be noted in the appropriate steps.

[@DATE\\_SEPARATOR](#)  
[@FATAL](#)  
[@HANGUP](#)  
[@PROMPT](#)  
[@SR\\_END\\_OF\\_VOICE\\_TIMEOUT](#)  
[@SYS\\_DATA](#)  
[@TIMEOUT](#)  
[@VOLUME](#)

#### 5.7.4.1 Application Parameters @P1 ... @P9

**Description:** The system variables @P1 through @P9 contain the parameter values passed in to the TeleFlow Application (.TAP) from the TeleFlow Application List file, or a Run Tap step. There are 9 fields in the [LineList](#) editor that allow you to pass values into your application. These fields can only contain literals.



If you use the [Run Tap step](#) to call another TeleFlow Application from within your current application, you can pass parameters to the second application in the appropriate fields in the Step Properties window for the Run Tap step. In this case, you can supply TeleFlow variables from your current application. They will be parsed and their values passed in to the second application.

**Data Type:** Any

#### 5.7.4.2 CALL\_ELAPSE

**Description:** The elapsed time, in seconds (with milliseconds precision) of the most recent call. If a call is in progress, this value will increase until the call is completed (Hang Up). Once a call has been completed, @CALL\_ELAPSE will retain the total duration of the call until a new call is started. Evaluates to 0.000 before the first call.

**Data Type:** Numeric (real number)

#### 5.7.4.3 CALL\_END

**Description:** Contains the end time of the most recent call, represented as the total number of seconds since midnight (with milliseconds precision). @CALL\_END evaluates to -1.000 before the first call, and while a call is active. @CALL\_END will retain its value until a new call has begun.

**Data Type:** Numeric (real number)

#### 5.7.4.4 CALL\_START

**Description:** Contains the start time of the current or most recent call, represented as the total number of seconds since midnight (with milliseconds precision). @CALL\_START will retain its value after a call is completed (Hang Up), but will be reset once a new call is started. @CALL\_START evaluates to -1.000 before the first call

**Data Type:** Numeric (real number)

#### 5.7.4.5 CALLOUT\_REASON

**Description:** A "code" indicating the reason for the @CALLOUT\_RESULT of the most recent outbound call made using the [Place Call](#) step. Consult the topic on [@CALLOUT\\_RESULT](#) for detailed information.

**Data Type:** String

#### 5.7.4.6 CALLOUT\_RESULT

**Description:** A "code" indicating the result of the most recent outbound call made using the [Place Call](#) step. The [@CALLOUT\\_REASON](#) system variable indicates why the @CALLOUT\_RESULT was what it was. The possible combinations of @CALLOUT\_RESULT and @CALLOUT\_REASON are listed below.

**Data Type:** String

Possible @CALLOUT\_RESULT and associated @CALLOUT\_REASON values (each table contains one CALLOUT\_RESULT and all the possible CALLOUT\_REASON values associated with it):

Variable = Code	Description
-----------------	-------------

@CALLOUT_RESULT = BUSY	The remote party was unreachable.
@CALLOUT_REASON = BUSY	The remote side is busy/engaged(i.e. Likely, the called party is on the phone).
@CALLOUT_REASON = REORDER	A reorder or fast-busy tone was detected. This is used to indicate an invalid dial string, or that all circuits (trunks) are busy and/or the call is unroutable.
@CALLOUT_REASON = REJECT	The remote side called rejected the call.

Variable = Code	Description
-----------------	-------------

@CALLOUT_RESULT = NOANSWER	The call wasn't answered in the time allowed by the Place Call step.
@CALLOUT_REASON = MAX_RINGS	The maximum ring count was reached without the remote party answering.

Variable = Code	Description
-----------------	-------------

@CALLOUT_RESULT = CONNECT	The call connected successfully.
@CALLOUT_REASON = VOICE	Voice was detected: Likely means that a person or automated system answered the phone at the remote side.
@CALLOUT_REASON = CED	A CED (fax tone) was detected, and the "Connect on CED" checkbox on the Place Call step was checked.
@CALLOUT_REASON = SIT	An SIT (special information tone) was detected, and the "Connect on SIT" checkbox on the Place Call step was checked.
@CALLOUT_REASON = CONNECT	A connection event, indicating that the call was successfully connected to the remote side, was received.

Variable = Code	Description
-----------------	-------------

@CALLOUT_RESULT = FAILURE	A failure generally indicates that something about the attempt to call went wrong, or that the number being called isn't available, and is unlikely to become so with a retry.
@CALLOUT_REASON = SIT	An SIT (special information tone) was detected, and the "Connect on SIT" checkbox on the Place Call step was not checked.
@CALLOUT_REASON = CED	A CED (fax tone) was detected, and the "Connect on CED" checkbox on the Place Call step was not checked.
@CALLOUT_REASON = NO_DIALTONE	Place call didn't detect a dial tone when it tried to go off-hook to place the call.

#### 5.7.4.7 CHAR\_ or Character variables

**Description:** TeleFlow creates a series of variables when it starts up that contain ASCII characters that are otherwise difficult or impossible generate in the TeleFlow Designer, but which might be required to serve some application needs. The @CHAR\_ variables are all named such that you can tell what the special character contents of the variables will be. For example: @CHAR\_TAB contains a single tab character.

Some of these are also available in legacy variables, such as [@LINEFEED](#).

**Data Type:** String

#### 5.7.4.8 DATE

**Description:** Contains the current system date, in [TeleFlow date format](#), YYYY/MM/DD (or YYYY-MM-DD - see notes below)

**Data Type:** Date

**Notes:** The [@DATE\\_SEPARATOR](#) system variable can be set to change the date separator to a dash, "-".

**See Also:**

[@DATETIME](#)  
[@DATETIME UTC](#)  
[@DATE\\_SEPARATOR](#)  
[@DATE UTC](#)  
[@TIME](#)  
[@TIME UTC](#)

#### 5.7.4.9 DATE\_SEPARATOR

**Description:** This setting variable controls the character used by TeleFlow to separate the portions of a TeleFlow date. You can set [@DATE\\_SEPARATOR](#) to either "-" or "/" (the default)

For legacy compatibility, the default setting is "/". For new applications it is recommended to change the date separator to a dash, "-", for compatibility with ISO 8601.

**Type:** String

**Notes:** Although you can set [@DATE\\_SEPARATOR](#) to any value, only "-" and "/" are valid. Any other setting will result in the default of "/" being used instead.

**See Also:**

[@DATETIME](#)  
[@DATETIME UTC](#)  
[@DATE](#)  
[@DATE UTC](#)

#### 5.7.4.10 DATE.UTC

**Description:** Whenever this variable is used, it returns the current UTC (Universal Coordinated Time) date. This value is in [TeleFlow date format](#), YYYY/MM/DD (or YYYY-MM-DD - see notes below).

**Data Type:** Date

**Notes:** The [@DATE\\_SEPARATOR](#) system variable can be set to change the date separator to a dash, "-".

**See Also:**

[@DATETIME](#)  
[@DATETIME UTC](#)  
[@DATE](#)  
[@DATE\\_SEPARATOR](#)  
[@TIME](#)  
[@TIME UTC](#)

#### 5.7.4.11 DATETIME

**Description:** Whenever this variable is used, it returns the current full date and time in [TeleFlow Date/Time Format](#), YYYY/MM/DD HH:MM:SS (or YYYY-MM-DD HH:MM:SS - see notes below).

**Type:** Date/Time

**Notes:** The [@DATE\\_SEPARATOR](#) system variable can be set to change the date separator to a dash, "-". This is strongly recommended for compatibility with ISO 8601.

**See Also:**

[@DATE](#)  
[@DATE\\_SEPARATOR](#)  
[@DATETIME UTC](#)  
[@DATE UTC](#)  
[@TIME](#)  
[@TIME UTC](#)

#### 5.7.4.12 DATETIME.UTC

**Description:** Whenever this variable is used, it returns the current time, adjusted to UTC (Universal Coordinated Time). The full date and time is returned in [TeleFlow Date/Time format](#), YYYY/MM/DD HH:MM:SS (or YYYY-MM-DD HH:MM:SS - see notes below).

**Type:** Date/Time.

**Notes:** The [@DATE\\_SEPARATOR](#) system variable can be set to change the date separator to a dash, "-". This is strongly recommended for compatibility with ISO 8601.

**See Also:**

[@DATETIME](#)  
[@DATE](#)  
[@DATE\\_SEPARATOR](#)  
[@DATE UTC](#)  
[@TIME](#)  
[@TIME UTC](#)

#### 5.7.4.13 FATAL

**Description:** The name of the label the application will go to when a [Fatal Error](#) occurs.

NOTE: You cannot set this variable directly with a Set Global or Set Local Variable step. Instead, use the [Global Event Step](#).

**Data Type:** String

#### 5.7.4.14 FORMFEED

**Description:** Contains a form feed character. (ASCII 12)

**Data Type:** String

#### 5.7.4.15 HANGUP

**Description:** The name of the label the application will go to when a caller hangs up.

NOTE: You cannot set this variable directly with a Set Global or Set Local Variable step. Instead, use the [Global Event Step](#).

**Data Type:** String

#### 5.7.4.16 HOOK\_STATUS

**Description:** Indicates whether the phone is on or off hook. If it is set to 'ON-HOOK', there is currently no call on the port, and steps/features requiring an active call (such as Play to play a file, or Get TT String to get touch tones from a caller) will fail (likely resulting in an application or fatal error). If it is set to 'OFF-HOOK', this means that there is a call active on the port.

**Data Type:** String

**Sample Uses:** This is helpful in flowcharts that are run both when a caller is on the line and when there is no one on the line. You might use this to determine whether or not to play a message to the caller. The most common example is playing a technical difficulties message in the event of an application error. When you go to the fatal error label, @HOOK\_STATUS is the only way to know if a caller is still on the line. If they are, you can play them a message. If not, and you attempt to play a message, this will generate a fatal error. If not handled carefully, this can create an endless loop.

This is also useful to determine what to do when returning to a parent TAP after using Run TAP to run a child or sub-application. When you leave another TAP, you may be leaving it because the functions of that application are completed, but the caller is still on the line interacting with your application. It may also be that a hang up was detected in the application, and you can use @HOOK\_STATUS to determine which of these is the case.

#### 5.7.4.17 LINEFEED

**Description:** Contains a carriage return character. (ASCII 13)

**Data Type:** String

#### 5.7.4.18 P1 to P9

**Description:** The system variables @P1 through @P9 contain the parameter values passed in to the TeleFlow Application (.TAP) from the TeleFlow Application List file, or a Run Tap step. There are 9 fields in the [LineList](#) editor that allow you to pass values into your application. These fields can only contain literals.

If you use the [Run Tap step](#) to call another TeleFlow Application from within your current application, you can pass parameters to the second application in the appropriate fields in the Step Properties window for the Run Tap step. In this case, you can supply TeleFlow variables from your current application. They will be parsed and their values passed in to the second application.

**Data Type:** Any

#### 5.7.4.19 PORT

**Description:** The port number the application is running on/for. (A single port is capable of carrying one "call") A port is also sometimes called a line (although in analog phone networks, a line often physically carries two individual calls) or channel (channel is a digital term). Only one TeleFlow application can manage a given port at one time. Which application(s) run on which port(s) is defined using the [TeleFlow Line List](#) utility.

**Data Type:** Numeric (whole number)

#### 5.7.4.20 PROMPT

**Description:** The path to the prompt files (pre-recorded audio files) to use when speaking data values through the [Say step](#). The @PROMPT system variable cannot be set directly, but can be set through the "Location of prompt recordings" property of the [Set Language](#) step. (This has not always been this way. If you have an old TeleFlow Application that sets @PROMPT directly, you need to update the application to use Set Language instead)

If it is not changed by the Set Language set, @PROMPT is set to the default: [INSTALLED DRIVE] : \Program Files\TeleFlow\Languages\English

**Data Type:** String

#### 5.7.4.21 RECORD\_RESULT

**Description:** A code indicating the result of the most recent [Record](#) step. Consult the following table for possible codes and their meanings.

@RECORD_RESULT	Description
SPEECH STOP	The recording completed because the caller stopped talking. (I.e. Silence TIMEOUT was reached at some point after a caller has started talking/recording)
NO VOICE	The recording completed because the caller never started talking. (I.e. Silence TIMEOUT was reached at the beginning of the recording)
MAX LENGTH	The recording completed because the maximum length of recording was reached. (I.e. The Maximum time recordable was exceeded)
INTERRUPTED	The recording completed because the caller pressed a key to interrupt it/complete the recording.

**Data Type:** String

#### 5.7.4.22 SPACE

**Description:** Contains a single space character. Although a space can be easily represented in a variable or field in TeleFlow Designer, it is much easier to see with the naked eye as a variable.

**Data Type:** String

#### 5.7.4.23 SPELLTONES

**Description:** This variable is used in conjunction with the [Get Letter step](#). @SPELLTONES will contain all of the touch-tones used to generate the last letter entered by a Get Letter step. This variable is cleared at the start of each new Get Letter Step.

**Data Type:** String

#### 5.7.4.24 SR\_END\_OF\_VOICE\_TIMEOUT

**Description:** In a Speech Recognition (SR) application, this is the length of silence time, expressed in milliseconds, required to terminate listening for new audio to recognize . Once this amount of time has elapsed in silence (the caller stops talking for this amount of time), the last word or words spoken by the caller are what the system uses (attempts to recognize) in the next speech recognition step.

If not set by an application, this variable defaults to 750 milliseconds.

**Data Type:** Numeric (whole number, milliseconds)

#### 5.7.4.25 SYS\_APP\_STATUS

**Description:** Returns the current application instance's running state. Possible values are:

**TF\_NORMAL** The application instance is running normally. This is the default state of an application instance when it is first started; when TeleFlow Server starts up, or when you perform a "Reload" operation using the [TeleFlow Monitor](#).

When an application instance is in this state, the status light displayed on the [TeleFlow Monitor](#) will be *bright green*.

**TF\_ERROR** The application instance has encountered a fatal error. Note that the error status will be indicated as soon as the error occurs, but application instance will continue to run for a short time, especially if you have a Fatal Error Handler defined.

When an application instance is in this state, the status light displayed on the [TeleFlow Monitor](#) will be *red*.

**TF\_RECOVERED** The application instance has recovered after previous fatal error. TeleFlow will automatically restart an application instance that terminated due to an error. For more information on this topic, refer to [Configure TeleFlow Server](#) or [TeleFlow LineList](#).

When an application instance is in this state, the status light displayed on the [TeleFlow Monitor](#) will be *yellow*.

**Data Type:** String

**Sample Uses:** Suppose an application calls a cell phone number to remind that person when they have a voicemail message on their office system. Due to a programming error in your application, the person called only hears part of the application before it terminates. When that application instance starts up again, you may want to know that it had previously errored so that you can ensure it does not call the same person again, in an endless loop.

**See Also:**

[Configure TeleFlow Server](#)  
[TeleFlow LineList](#)  
[TeleFlow Monitor](#)

#### 5.7.4.26 SYS\_CALLREF

**Description:** Useable only on ISDN ports, this variable contains the system call reference number for the current call. As long as the port is on a call, the variable will contain this value. If the port is on-hook, or the port is not an ISDN port, the variable will be empty.

**Data Type:** Numeric (whole number)

**Sample Uses:** The call reference value of a port is required to perform a 2B-Channel transfer. See the [Transfer](#) step for more information.

#### 5.7.4.27 SYS\_DATA

**Description:** The contents of this variable are displayed in the "Data" column of the [TeleFlow Monitor](#). It can be set in an application at any time with a [Set Global Variable](#) step.

**Data Type:** String

#### 5.7.4.28 SYSDIR

**Description:** Contains the directory/folder location - without the drive - of the running .TAP file. For example, if you were running C:\YourApplication\NewApp.TAP, @SYSDIR would be: '\YourApplication'.

**Data Type:** String

#### 5.7.4.29 SYSDRIVE

**Description:** Contains the drive designation of the running TAP file. For example, if you were running your application from C:\YourApplication, @SYSDRIVE would return:"C:".

**Data Type:** String

#### 5.7.4.30 TIME

**Description:** Whenever this variable is used, it returns the current time-of-day, in [TeleFlow time format](#), HH:MM:SS.

**Data Type:** Time

**Notes:** Only the time-of-day is returned. To get the full date and time value, use [@DATETIME](#).

**See Also:**

[@DATETIME](#)  
[@DATETIME UTC](#)  
[@DATE](#)  
[@DATE UTC](#)  
[@TIME UTC](#)

#### 5.7.4.31 TIME.UTC

**Description:** Whenever this variable is used, it returns the current time-of-day, adjusted to UTC (Universal Coordinated Time), in [TeleFlow time format](#), HH:MM:SS.

**Data Type:** Time

**Notes:** Only the time-of-day is returned. To get the full date and time value, use [@DATETIME.UTC](#).

**See Also:**

[@DATETIME](#)  
[@DATETIME UTC](#)  
[@DATE](#)  
[@DATE UTC](#)  
[@TIME](#)



#### 5.7.4.32 TIMEOUT

**Description:** This variable is a system wide [Time-Out](#) default - expressed in milliseconds - for all steps that require a timeout value (e.g.: [Get TT String](#) Step). It can be set in a TeleFlow application with the [Set Global Variable](#) step. If the application doesn't set @TIMEOUT, it defaults to 3000 milliseconds, or 3 seconds.

On any step that has a field label that contains "TIMEOUT", the field will use @TIMEOUT if the field is not specifically filled in with another value.

Steps with timeouts keep track of how long the step has been running without any activity from the caller. When the step doesn't get any activity for the timeout value supplied it will follow its [Time-out Action](#) path. How this works with a given step varies based on the needs of the step.

**Data Type:** Numeric (whole number, milliseconds)

#### 5.7.4.33 TRUNK\_TYPE

**Description:** Contains a single word indicating the type of trunk the application's port is associated with. (I.e.: analog, digital T1, etc.) @TRUNK\_TYPE will contain one of the following values:

NONE: Application is not using a port.

SIMULATED: Application is running on the TeleFlow Simulator

T1: Application is using a port on a digital T1 trunk. Regardless of the configuration of the trunk, the value will still be "T1"

E1: Application is using a port on a digital E1 trunk. Regardless of the configuration of the trunk, the value will still be "E1"

ANALOG: Application is using an analog port. Regardless of the configuration of the analog trunk (eg: loop-start, ground-start), the value will still be "ANALOG"

VOIP: Application is using a voice-over-IP (VoIP) port.

**Data Type:** String

**Sample Uses:** Sometimes it is important to know what kind of trunk your application is running on, so you can use a [compare](#) step to check the trunk type before you execute functionality that is only available on certain types. For example, on some analog telephone circuits, you can perform a flash-hook to put the caller on hold. The same functionality may not be available if you run your application on a digital T1 port.

#### 5.7.4.34 UNIX\_TIME

**Description:** Whenever you use this read-only system variable, it will return the current system time in the format made popular by the Unix operating system (although this format is used widely in the information technology sector).

The value returned is the total number of seconds since Midnight, December 31, 1969, including fractional seconds to millisecond precision (1/1000 of a second).

Example: 1128039196.552

**Data Type:** Numeric (real number)

**Sample Uses:** This time format is extremely useful for performing math operations on two different times. For example, you could calculate the time difference between two events. To do so, capture the time at the starting event by setting a variable to `@UNIX_TIME`:

[Set Local Variable](#) `@START_TIME = @UNIX_TIME`

Later in your application you can calculate the difference between the current time, and the previously captured time with a single [Math step](#) as follows:

[Math](#) `@TIME_DIFF = @UNIX_TIME - @START_TIME`

#### 5.7.4.35 UTC\_DATE

**Description:** Synonym for [@DATE.UTC](#).

**Type:** Date

**Notes:** This is a deprecated name; use [@DATE.UTC](#) instead.

#### 5.7.4.36 UTC\_TIME

**Description:** Synonym for [@TIME.UTC](#).

**Notes:** This is a deprecated name; use [@TIME.UTC](#) instead.

**Type:** Time

#### 5.7.4.37 VOLUME

**Description:** Current volume setting, as set by either the [Set Volume](#) step, or directly using a [Set Global Variable](#) step, used when playing prompts, messages, and for recording messages. The default value of zero indicates prompts should play at the same volume level they were recorded with. The possible range of values is from -3 (quietest) to 3 (loudest).

**Data Type:** Numeric (whole number, -3 to 3)

**Sample Uses:** In a voicemail system, it can be helpful to record voice messages louder than scripts and prompts are played. Some callers are fairly quiet, and louder recordings can help when listening to voicemail. To take advantage of the `@VOLUME` variable, you can save `@VOLUME` just prior to a record step by using a Set Global or Set Local variable step to capture the current value (Eg. `@PLAY_VOLUME = @VOLUME`). Then, set `@VOLUME` to a new, louder setting. After the record step, set `@VOLUME` back to the previous value using Set Global Variable (Eg. `@VOLUME = @PLAY_VOLUME`)

Alternatively, you can have two global variables, such as `@PLAY_VOLUME` and `@RECORD_VOLUME`, and set `@VOLUME` equal to whichever is appropriate for the application as the time of the setting.

#### 5.7.4.38 WEEKDAY

**Description:** Contains the first 3 letters of the current weekday name in uppercase. (So, valid values for this variable are: MON, TUE, WED, THU, FRI, SAT, SUN) For additional weekday and date variable manipulation, see the [@DATE](#) system variable, and the [Date Info](#) step.

**Data Type:** String

**Sample Uses:** Use this variable in a [Compare](#) step to decide whether or not to place an outbound call. If you have specific business rules for the time of day in which to place calls on the weekend, a couple of Compare (or a single [Case](#) step) steps to check for 'SAT' or 'SUN', followed by a check for the current time using the [@TIME](#) system variable should do the trick.

## 5.8 Voice Script Manager

[What is the Voice Script Manager?](#)

[Getting Around in Voice Scripts Manager](#)

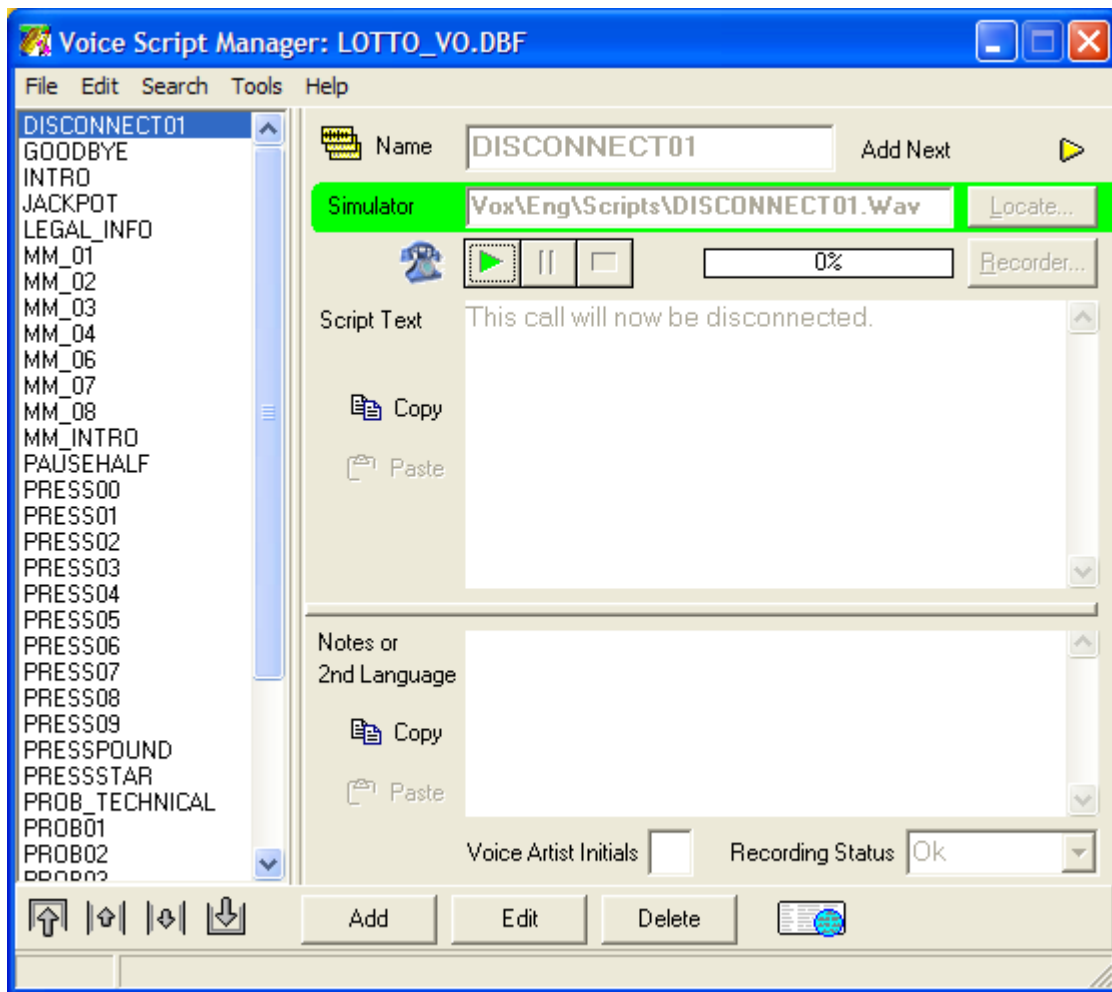
[Adding Voice Scripts to an Application](#)

[Recording Voice Files with the Script Manager](#)

[Using Existing or Pre-Recorded Voice Files](#)

[Editing Voice Scripts](#)

### 5.8.1 What is the Voice Script Manager?



A completed IVR application may contain hundreds or even thousands of different voice files. The Voice Script Manager is an easy to use menu system that allows the developer to record, delete, or edit voice files. There are several ways to access the Voice Script Manager:

- using the Hot buttons on the Step properties boxes for certain Step Icons (play, say, etc)
- using the Voice Script icon on the TAP Screen

As an application is built, you should type in the scripts via the Script Manager as you place the step icons on the flowcharts. This ensures that you will not forget what each step icon is supposed to be playing. Also, by typing in the scripts as you go, the Designer can automatically print you a complete list of scripts when it comes time to record the actual voice files. The Script Manager window has two colored areas on the screen. Their functions are:

- Yellow - File Location and type of voice file (for the actual application to run)
- Cyan - Location of files for the [TeleFlow Simulator](#) and a recording studio so that files can actually be recorded on the fly if you have a microphone handy

Contents:

- [Adding Voice Scripts to an Application](#)
- [Editing Voice Scripts](#)
- [Managing Repetitive Voice Scripts](#)
- [Recording Voice Files with the Script Manager](#)
- [Using Existing or Pre-Recorded Voice Files](#)

## 5.8.2 Getting Around in Voice Scripts Manager

Voice Script Manager Menus:

[Edit](#)  
[File](#)  
[Help](#)  
[Tools](#)  
[Window](#)

### 5.8.2.1 Edit Menu

**Copy Script into Clipboard Buffer** **Ctrl+F1**

Copies the text in the script section into the clipboard so that it can be pasted elsewhere.

**Paste Clipboard Buffer into Script** **Ctrl+F2**

Pastes the information currently in the clipboard to the script section.

**Copy Notes into Clipboard Buffer** **Ctrl+F3**

Copies the text in the note section into the clipboard so that it can be pasted elsewhere.

**Paste Clipboard Buffer into Notes** **Ctrl+F4**

Pastes the information currently in the clipboard to the note section.

### 5.8.2.2 File Menu

**Print**

This will print the scripts for all of the voice scripts that have been entered through the Voice Script Manager. This is normally done in preparation for going to the recording studio that will create the voice files.

**Exit**

This exits the Voice Script Manager and takes you back to the previous window.

### 5.8.2.3 Help Menu

**Contents**

This will take you into the help file, and will open it at the page specifically on the [Voice Script Manager](#).

#### 5.8.2.4 Tools Menu

##### Auto Play

Auto play is either checked on or off, and, if on, will automatically play any recorded files that are associated with a script that you edit in the Voice Script Manager.

#### 5.8.2.5 Windows Menu

##### Spreadsheet

This command will open a spreadsheet window that lays out all of the voice scripts and their attributes into a graph format. As you click on each script filename, the voice script manager will automatically update its screen with the appropriate information. You can change the filename from the spreadsheet also.

### 5.8.3 Adding Voice Scripts to an Application

In the Voice Script Manager, create the name of the voice script (i.e. PRESS01), add the text for the voice file that you are adding into the script box near the bottom of the window, and then click on the 'locate' button that is in the yellow highlighted area. A window similar to a standard 'BROWSE' window will open, and you should navigate to the directory that contains your existing voice file. Double click on the filename, and it will now appear in the Telephony box just to the left of the 'locate' button. If you wish this voice file to be used when you run the [TeleFlow Simulator](#), then repeat the procedure with the 'locate' button that is in the blue highlighted area.

NOTE: It is not necessary to create your voice files using the recorder that is built into TeleFlow Designer. If you wish to use a different program to record your voice files, you may do so. Physically, there is no difference between the file formats that are recorded, but a program that is specifically used for playing and recording sound files may offer you more options. Ensure that you are recording in a standard wave (.WAV) format, and put the recorded files into a specific voice file directory (if you keep all of your voice files in one separate directory, it will make it easier if you need to find them at a later time).

### 5.8.4 Recording Voice Files with the Script Manager

The commands to record and/or review voice files are found in the [Voice Script Manager](#). All associated command buttons are highlighted blue, and they include play, pause, stop, and record. There is also a button for locating existing voice files on the hard drive (much like a 'browse'), and a 'recorder' button which will load an external recording program of your choice.

As you are creating voice scripts, start with recording scripts for use with the [TeleFlow Simulator](#). After you have typed in the text of a [voice script](#) you can record the voice file by clicking on the 'REC' button. The play and record buttons will become active. If you are creating a new recording, click on the record button, and speak the content of the text into the microphone. Once recording, the 'stop' button is activated so that you can click on it when you are done recording. To hear the results, click on the 'play' button.

**NOTE:** If you need to record a voice file again, you must click on the 'Reset' button before clicking on the record button. Clicking on the 'Reset' button first will reset that particular voice file. The record button will only add what is recorded to the end of the existing file, so in order to record over the existing file it must be reset first.

Accordingly, if you wish to add more to the end of an existing voice file, you may do this by simply clicking on the record button, and speaking your addition into the microphone.

The recording feature is only appropriate for smaller applications. If the IVR system being developed is going to change often or be customized from client to client, you may want to consider using other third party tools for voice recording management. We highly recommend Syntrillium's Cool Edit. (A

link to this can be found in the 'Third Party Links' section of the Help) The evaluation version of this software can be found on the TeleFlow Installation CD.

### 5.8.5 Using Existing or Pre-Recorded Voice Files

It is not necessary to create your voice files using the recorder that is built into TeleFlow Designer. If you wish to use a different program to record your voice files, you may do so. Physically, there is no difference between the file formats that are recorded, but a program that is specifically used for playing and recording sound files may offer you more options. Ensure that you are recording in a standard wave (.WAV) format, and put the recorded files into a specific voice file directory (if you keep all of your voice files in one separate directory, it will make it easier if you need to find them at a later time).

In the [Voice Script Manager](#), create the name of the voice script (i.e. PRESS01), add the text for the voice file that you are adding into the script box near the bottom of the window, and then click on the 'locate' button that is in the yellow highlighted area. A window similar to a standard 'BROWSE' window will open, and you should navigate to the directory that contains your existing voice file. Double click on the filename, and it will now appear in the Telephony box just to the left of the 'locate' button.

If you wish this voice file to be used when you run the [TeleFlow Simulator](#), then repeat the procedure with the 'locate' button that is in the blue highlighted area.

### 5.8.6 Editing Voice Scripts

The Scripts Manager is essentially a database of all the text segments that comprise an application. This screen also includes a simple wave file recorder to record 8000Hz 16bit PCM wave files that will work with the [TeleFlow Simulator](#). Adding and editing new scripts is an easy process using the Voice Script Manager. Simply enter into the Manager via the TAP screen or use the hot button on the step property box of any dialogue icon (play, say, etc). Using the forward/backward buttons or the pull down menu in the reference box, find the voice script that you want to edit, and then click on the modify button (triangle point up). Simply type the changes that you want into the 'Script' box, and, if you wish, re-record the changes to the voice file using the record functions.

## 5.9 Voice Prompts

When using TeleFlow's [Say](#) step to play data, TeleFlow plays pre-recorded prompt files, putting together the files necessary to say the data in the step appropriately. To play the number 150, for example, would require the files one.wav, hundred.wav, and fifty.wav.

The topics in this chapter provide the complete word set(s) required to enable TeleFlow to string words together to play data in the Say step.

[English Standard Word Set](#)  
[English to French Standard Word Set](#)

### 5.9.1 English Standard Word Set

This word set is used by TeleFlow to build phrases such as times, dates, and numbers. Messages from 100 to 159 are used to spell words and text to the listener. Using this method, different messages are added together to speak a single piece of information.

Examples:

- \$1112.08 would convert to "One" "Thousand" "One" "Hundred" "Twelve" "Dollars" "And" "Eight" "Cents"
- 1997/11/21 would convert to "November" "Twenty" "First" "Nineteen" "Ninety" "Seven"

**Voice Artist and Sound Engineer Instructions:**

Since full phrases are built from partial phrases, care must be taken to ensure that each word phrase is spoken distinctly and separate from another. The same tone and consistency must be kept from word to word in order to make the above structured sentences sound proper.

Before recording this section imagine how your voice would need to sound in order for the computer system to piece together these kinds of sentences. Speaking each word as a statement and not as a question will help with the process. Be careful to not to rattle off this section as if it were a sentence. It will sound appropriate if a three-second space is left between each word. Speaking in *Staccato* can be a good warm up for this section.

Finally, try to keep each word sounding as if it were spoken in a conversational manner. For example, speak "Thousand" like "thousand", and not like "th-oooww-sand"

## WORD SET

The following sections each represent a subset of the full English Standard Word Set. It has been split up so that recordings can be done according to what a system is likely to require, rather than having to record everything, even when it won't be used.

The first, or 'Mandatory' section, consists of numbers, date, time and monetary words.

The second, or 'Alphabet' section, is made up of the letters of the alphabet only.

The third and final, or 'Special Characters' section, contains punctuation and other special characters, such as the 'percent' sign or 'greater than' sign.

Each section is laid out in columns. The leftmost column is simply a row/script number. The middle is the word or words that should be spoken for that particular script. The rightmost column is the name the recorded file must be save to in order for the TeleFlow Server to use it when necessary.

### MANDATORY SECTION

MESSAGE #	VOICE SCRIPT	WAVE FILE
01	Zero	00.wav
02	One	01.wav
03	Two	02.wav
04	Three	03.wav
05	Four	04.wav
06	Five	05.wav
07	Six	06.wav
08	Seven	07.wav
09	Eight	08.wav
10	Nine	09.wav
11	Ten	10.wav
12	Eleven	11.wav
13	Twelve	12.wav
14	Thirteen	13.wav
15	Fourteen	14.wav
16	Fifteen	15.wav
17	Sixteen	16.wav
18	Seventeen	17.wav
19	Eighteen	18.wav



20	Nineteen	19.wav
21	Twenty	20.wav
22	Thirty	30.wav
23	Forty	40.wav
24	Fifty	50.wav
25	Sixty	60.wav
26	Seventy	70.wav
27	Eighty	80.wav
28	Ninety	90.wav
29	Hundred	Hundred.wav
30	Thousand	Thousand.wav
31	Million	Million.wav
32	Billion	Billion.wav
33	Trillion	Trillion.wav
34	January	JAN.wav
35	February	FEB.wav
36	March	MAR.wav
37	April	APR.wav
38	May	MAY.wav
39	June	JUN.wav
40	July	JUL.wav
41	August	AUG.wav
42	September	SEP.wav
43	October	OCT.wav
44	November	NOV.wav
45	December	DEC.wav
46	Seconds	Seconds.wav
47	Minus	Minus.wav
48	Dollar	Dollar.wav
49	Dollars	Dollars.wav
50	Cent	Cent.wav
51	Cents	Cents.wav
52	And	And.wav
53	First	1st.wav
54	Second	2nd.wav
55	Third	3rd.wav
56	Fourth	4th.wav
57	Fifth	5th.wav
58	Sixth	6th.wav
59	Seventh	7th.wav
60	Eighth	8th.wav
61	Ninth	9th.wav
62	Tenth	10th.wav
63	Eleventh	11th.wav
64	Twelfth	12th.wav

65	Thirteenth	13th.wav
66	Fourteenth	14th.wav
67	Fifteenth	15th.wav
68	Sixteenth	16th.wav
69	Seventeenth	17th.wav
70	Eighteenth	18th.wav
71	Nineteenth	19th.wav
72	Twentieth	20th.wav
73	Thirtieth	30th.wav
74	[0.4 second pause]	Pause.wav
75	O'clock	Oclock.wav
76	A.M	AM.wav
77	P.M	PM.wav
78	Oh	Oh.wav
79	Point	Point.wav
80	Monday	Mon.wav
81	Tuesday	Tue.wav
82	Wednesday	Wed.wav
83	Thursday	Thu.wav
84	Friday	Fri.wav
85	Saturday	Sat.wav
86	Sunday	Sun.wav
87		
88		
89		
90		
91		
92		
93		
94		
95		
96		
97		
98		
99	Hours	Hours.wav
100	Hour	Hour.wav

OPTIONAL SECTION 1, "ALPHABET"

MESSAGE #	VOICE SCRIPT	WAVE FILE
101	A	A.wav
102	B	B.wav
103	C	C.wav
104	D	D.wav

105	E	E.wav
106	F	F.wav
107	G	G.wav
108	H	H.wav
109	I	I.wav
110	J	J.wav
111	K	K.wav
112	L	L.wav
113	M	M.wav
114	N	N.wav
115	O	O.wav
116	P	P.wav
117	Q	Q.wav
118	R	R.wav
119	S	S.wav
120	T	T.wav
121	U	U.wav
122	V	V.wav
123	W	W.wav
124	X	X.wav
125	Y	Y.wav
126	ZED	ZED.wav
127	ZEE	ZEE.wav

#### OPTIONAL SECTION 2: "SPECIAL CHARACTERS"

<b>MESSAGE #</b>	<b>VOICE SCRIPT</b>	<b>WAVE FILE</b>
128	Exclamation Point	exclamation. wav
129	Double quote	dquote.wav
130	Pound sign	pound.wav
131	Dollar sign	dsign.wav
132	Percent sign	percent.wav
133	Ampersand	ampersand.w av
134	Single quote	squote.wav
135	Open parentheses	openparen.w av
136	Closed parentheses	closparen.wa v
137	Asterisk	asterisk.wav
138	Plus sign	plus.wav
139	Comma	comma.wav
140	Hyphen	hyphen.wav
141	Period	period.wav

142	Forward slash	fslash.wav
143	Colon	colon.wav
144	Semicolon	semicolon.wav
145	Less than symbol	lessthan.wav
146	Equal sign	equal.wav
147	Greater than symbol	morethanv
148	Question mark	question.wav
149	At sign	atsign.wav
150	Open square bracket	lsqbracket.wav
151	Back slash	bslash.wav
152	Close square bracket	rsqbracket.wav
153	Caret	caret.wav
154	Underscore	underscore.wav
155	Grave accent	grave.wav
156	Open brace	lbrace.wav
157	Vertical bar (line)	vertbar.wav
158	Close brace	rbrace.wav
159	Tilde	tilde.wav

### 5.9.2 English to French Standard Word Set

This word set is used by TeleFlow to build phrases such as times, dates, and numbers. Messages from 100 to 159 are used to spell words and text to the listener. Using this method, different messages are added together to speak a single piece of information.

Examples:

- \$1112.08 would convert to "One" "Thousand" "One" "Hundred" "Twelve" "Dollars" "And" "Eight" "Cents"
- 1997/11/21 would convert to "November" "Twenty" "First" "Nineteen" "Ninety" "Seven"

#### Voice Artist and Sound Engineer Instructions:

Since full phrases are built from partial phrases, care must be taken to ensure that each word phrase is spoken distinctly and separate from another. The same tone and consistency must be kept from word to word in order to make the above structured sentences sound proper.

Before recording this section imagine how your voice would need to sound in order for the computer system to piece together these kinds of sentences. Speaking each word as a statement and not as a question will help with the process. Be careful to not to rattle off this section as if it were a sentence. It will sound appropriate if a three-second space is left between each word. Speaking in *Staccato* can be a good warm up for this section.

Finally, try to keep each word sounding as if it were spoken in a conversational manner. For example, speak "Thousand" like "thousand", and not like "th-oooww-sand"

#### WORD SET

The following sections each represent a subset of the full French Standard Word Set. It has been split up so that recordings can be done according to what a system is likely to require, rather than having to record everything, even when it won't be used.

The first, or 'Mandatory' section, consists of numbers, date, time and monetary words.

The second, or 'Alphabet' section, is made up of the letters of the alphabet only.

The third and final, or 'Special Characters' section, contains punctuation and other special characters, such as the 'percent' sign or 'greater than' sign.

Each section is laid out in columns. The leftmost column is simply a row/script number. The middle is the word or words that should be spoken for that particular script. The rightmost column is the name the recorded file must be save to in order for the TeleFlow Server to use it when necessary.

#### MANDATORY SECTION

<b>MESSAGE #</b>	<b>ENGLISH SCRIPT</b>	<b>FRENCH SCRIPT</b>	<b>WAVE FILE</b>
01	Zero	Zero	00.wav
02	One	Un	01.wav
03	Two	Deux	02.wav
04	Three	Trois	03.wav
05	Four	Quatre	04.wav
06	Five	Cinq	05.wav
07	Six	Six	06.wav
08	Seven	Sept	07.wav
09	Eight	Huit	08.wav
10	Nine	Neuf	09.wav
11	Ten	Dix	10.wav
12	Eleven	Onze	11.wav
13	Twelve	Douze	12.wav
14	Thirteen	Treize	13.wav
15	Fourteen	Quatorze	14.wav
16	Fifteen	Quinze	15.wav
17	Sixteen	Seize	16.wav
18	Seventeen	dix-sept	17.wav
19	Eighteen	dix-huit	18.wav
20	Nineteen	dix-neuf	19.wav
21	Twenty	Vingt	20.wav
22	Thirty	Trente	30.wav
23	Forty	Quarante	40.wav
24	Fifty	Cinquante	50.wav
25	Sixty	Soixante	60.wav
26	Seventy	Soixante-dix	70.wav
27	Eighty	Quatre-vingt	80.wav
28	Ninety	Quatre-vingt-dix	90.wav

29	Hundred	Cent	Hundred.wav
30	Thousand	Mille	Thousand.wav
31	Million	Million	Million.wav
32	Billion	Billion	Billion.wav
33	Trillion	Trillion	Trillion.wav
34	January	Janvier	JAN.wav
35	February	Fevrier	FEB.wav
36	March	Mars	MAR.wav
37	April	Avril	APR.wav
38	May	Mai	MAY.wav
39	June	Juin	JUN.wav
40	July	Juillet	JUL.wav
41	August	Aout	AUG.wav
42	September	Septembre	SEP.wav
43	October	Octobre	OCT.wav
44	November	Novembre	NOV.wav
45	December	Decembre	DEC.wav
46	Seconds	Secondes	Seconds.wav
47	Minus	Moins	Minus.wav
48	Dollar	Dollar	Dollar.wav
49	Dollars	Dollar	Dollars.wav
50	Cent	Sous	Cent.wav
51	Cents	Sous	Cents.wav
52	And	Et	And.wav
53	First	Premier	1st.wav
54	Second	Deuxieme	2nd.wav
55	Third	Troisieme	3rd.wav
56	Fourth	Quatrieme	4th.wav
57	Fifth	Cinquieme	5th.wav
58	Sixth	Sixieme	6th.wav
59	Seventh	Septieme	7th.wav
60	Eighth	Huitieme	8th.wav
61	Ninth	Neuvieme	9th.wav
62	Tenth	Dixieme	10th.wav
63	Eleventh	Onzieme	11th.wav
64	Twelfth	Douzieme	12th.wav
65	Thirteenth	Treizieme	13th.wav
66	Fourteenth	Quatorzieme	14th.wav
67	Fifteenth	Seizieme	15th.wav
68	Sixteenth	Seizieme	16th.wav
69	Seventeenth	Dix-septieme	17th.wav
70	Eighteenth	Dix-huitieme	18th.wav
71	Nineteenth	Dix-neuvieme	19th.wav
72	Twentieth	Vingtieme	20th.wav
73	Thirtieth	Trentieme	30th.wav

74	[0.4 second pause]	[0.4 second pause]	Pause.wav
75	O'clock	Heure	O'clock.wav
76	A.M	Avant-midi	AM.wav
77	P.M	Après-midi	PM.wav
78	Oh	Zero	Oh.wav
79	Point	Point	Point.wav
80	Monday	Lundi	Mon.wav
81	Tuesday	Mardi	Tue.wav
82	Wednesday	Mercredi	Wed.wav
83	Thursday	Jeudi	Thu.wav
84	Friday	Vendredi	Fri.wav
85	Saturday	Samedi	Sat.wav
86	Sunday	Dimanche	Sun.wav
87	At	A	At.wav
88			
89			
90			
91			
92			
93			
94			
95			
96			
97			
98			
99	Hours	Hours	Hours.wav
100	Hour	Hour	Hour.wav

OPTIONAL SECTION 1, "ALPHABET"

MESSAGE #	VOICE SCRIPT	WAVE FILE
101	A	A.wav
102	B	B.wav
103	C	C.wav
104	D	D.wav
105	E	E.wav
106	F	F.wav
107	G	G.wav
108	H	H.wav
109	I	I.wav
110	J	J.wav
111	K	K.wav
112	L	L.wav

113	M	M.wav
114	N	N.wav
115	O	O.wav
116	P	P.wav
117	Q	Q.wav
118	R	R.wav
119	S	S.wav
120	T	T.wav
121	U	U.wav
122	V	V.wav
123	W	W.wav
124	X	X.wav
125	Y	Y.wav
126	ZED	ZED.wav
127	ZEE	ZEE.wav

OPTIONAL SECTION 2: "SPECIAL CHARACTERS"

<b>MESSAGE #</b>	<b>VOICE SCRIPT</b>	<b>WAVE FILE</b>
128	Exclamation Point	exclamation.wav
129	Double quote	dquote.wav
130	Pound sign	pound.wav
131	Dollar sign	dsign.wav
132	Percent sign	percent.wav
133	Ampersand	ampersand.wav
134	Single quote	squote.wav
135	Open parentheses	openparen.wav
136	Closed parentheses	closparen.wav
137	Asterisk	asterisk.wav
138	Plus sign	plus.wav
139	Comma	comma.wav
140	Hyphen	hyphen.wav
141	Period	period.wav
142	Forward slash	fslash.wav
143	Colon	colon.wav
144	Semicolon	semicolon.wav
145	Less than symbol	lessthan.wav
146	Equal sign	equal.wav
147	Greater than symbol	morethanv
148	Question mark	question.wav
149	At sign	atsign.wav
150	Open square bracket	lsqbracket.wav
151	Back slash	bslash.wav
152	Close square bracket	rsqbracket.wav



153	Caret	caret.wav
154	Underscore	underscore.wav
155	Grave accent	grave.wav
156	Open brace	lbrace.wav
157	Vertical bar (line)	vertbar.wav
158	Close brace	rbrace.wav
159	Tilde	tilde.wav

**TeleFlow**

**Part**



## 6 2 - Test Applications with TeleFlow Simulator



With this on-screen engine, you can test functions of any application without the need for a telephone. By using the multi media capabilities of your laptop or desktop computer, you can build a fully functional application and know that it will work once you install it on a voice board equipped IVR computer.

The toolbar section above the flowchart provides the functions to start Simulator, and pass parameters to it.

By clicking on the Simulator icon from the Designer, the current application will be automatically saved and loaded by the Simulator. Optionally, you can access Simulator from the 'Tools' menu. Simulator will then appear with a stylized telephone keypad. Once the application is loaded, it will automatically start running, carrying out the instructions as indicated in the flowcharts. Application progress can be reviewed by watching the messages window located below the keypad.

Beside the telephone buttons is a telephone icon, which is used to indicate whether the phone is off the hook or not. Click on the Call System button to hang up or answer the phone if the application permits. At times, this option will not be available because the application is not in the process of conducting a call. When this is the case, the phone may appear "off hook," but the Simulator will not run past "Waiting for Call" until you hang up and "Call System" again.

The Pause/Continue button below the telephone icon can be used to stop the progress of the application for demonstration and review purposes. Voice prompts will continue to play until finished, and the Simulator will pause at the next step.

**The following options are available from the Simulator menu:**

### File

**Open**

Open TeleFlow application

	<b>Set Log Directory</b>	Where Simulator stores log files
	<b>Exit</b>	Exit Simulator.
<b>Simulator</b>		
	<b>Start</b>	Start or restart the Simulator
	<b>Resume</b>	Continue operation after pausing Simulator
	<b>Stop</b>	Stop Simulator
<b>Settings</b>		
	<b>Log Directory</b>	Where Simulator stores log files
	<b>Maximum Log Size</b>	Set the maximum size of the log file
	<b>Short Audio Prompts</b>	Will only play
	<b>Freeze Display at Error</b>	Will hold the Simulator display at the error which caused Simulator to halt
<b>Help</b>		
	<b>About</b>	Version of Simulator.

**TeleFlow**

**Part**



## 7 3 - Configure TeleFlow Server

To run a TeleFlow Application live (or to test one on a phone system), you must configure TeleFlow Server.

The following topics explain how to use the server utilities provided to configure your TeleFlow Server.

[TeleFlow LineList](#): Creates TeleFlow Application List (TAL) files. A list file defines which TeleFlow Application's (TAP's) to run on which lines.

[TeleFlow Config](#): Configures the TeleFlow Server runtime settings.

[TeleFlow Register](#): TeleFlow Register is used to enter the registration information required to activate TeleFlow licenses.

### 7.1 Development or Production Options

[Installing TeleFlow Server](#)

### 7.2 TeleFlow Line List

The TeleFlow Line List utility is for building **.TAL - TeleFlow Application List** - files. An 'active' TAL file is read in by the [TeleFlow Server](#) each time it is started, and for each 'line' in the TAL file, the Server starts an application 'instance' configured as per the data for the line. There can be only one 'active' TAL file on a given computer at any one time, and this is specified in [TeleFlow Config](#).

Using the following screen capture as an example, the first 'line' in the TAL file is currently selected (The row starting with 'inbound01'). Typically, you will have one application service multiple callers, with each instance interacting with one caller. The instance named 'inbound01' would run a TAP called 'ANSWERTEST.TAP', which will answer an incoming call on port 1. When a call is answered, the application proceeds to interact with the caller: playing voice files, getting responses (touch tones) from the caller, etc. In order to have other callers get the same service as the caller on Port 1, ANSWERTEST.TAP must run on additional ports. The example has done just that: The ANSWERTEST.TAP is set up to run on 8 ports.

All of this assumes that there are the correct number of phone lines connected to your hardware (voice/telephony cards), and that everything is configured appropriately. To build an application list, consult the 'Building a TAL file' and 'Application List Fields' sections.

The screenshot shows a window titled "TeleFlow Line List - C:\TestTaps\Default.tal". The window contains a table with the following columns: Name, Filename, Protocol, Restart Count, No limit, Log output, Append to file, @PORT, @P1, @P2, and @P3. The table lists 8 inbound lines, each with a unique name (inbound01 to inbound08), the same filename (C:\TestTaps\ANSWERTEST.TAP), and the same protocol (Ips0). The Restart Count is 0 for all. The 'No limit' checkbox is checked for lines inbound03 through inbound08. The 'Log output' is set to 'All events' for inbound01 and 'Errors only' for the others. The 'Append to file' checkbox is checked for lines inbound03 through inbound08. The @PORT column shows values 1 through 8, while @P1, @P2, and @P3 are empty.

Name	Filename	Protocol	Restart Count	No limit	Log output	Append to file	@PORT	@P1	@P2	@P3
inbound01	C:\TestTaps\ANSWERTEST.TAP	Ips0	0	<input type="checkbox"/>	All events	<input type="checkbox"/>	1			
inbound02	C:\TestTaps\ANSWERTEST.TAP	Ips0	5	<input type="checkbox"/>	Errors only	<input type="checkbox"/>	2			
inbound03	C:\TestTaps\ANSWERTEST.TAP	Ips0	0	<input checked="" type="checkbox"/>	Errors only	<input checked="" type="checkbox"/>	3			
inbound04	C:\TestTaps\ANSWERTEST.TAP	Ips0	0	<input checked="" type="checkbox"/>	Errors only	<input checked="" type="checkbox"/>	4			
inbound05	C:\TestTaps\ANSWERTEST.TAP	Ips0	0	<input checked="" type="checkbox"/>	Errors only	<input checked="" type="checkbox"/>	5			
inbound06	C:\TestTaps\ANSWERTEST.TAP	Ips0	0	<input checked="" type="checkbox"/>	Errors only	<input checked="" type="checkbox"/>	6			
inbound07	C:\TestTaps\ANSWERTEST.TAP	Ips0	0	<input checked="" type="checkbox"/>	Errors only	<input checked="" type="checkbox"/>	7			
inbound08	C:\TestTaps\ANSWERTEST.TAP	Ips0	0	<input checked="" type="checkbox"/>	Errors only	<input checked="" type="checkbox"/>	8			

## Building a TAL file

TeleFlow Line List reduces the amount of work you have to do when building a TAL file. To create the sample above, only 1 line of information was filled in. It was then replicated to fill in the remaining 7 rows.

Consulting the 'Application List Fields' section below, fill in a single line of information for the TeleFlow Application you wish to run. Once you have done this, click 'Edit', select 'Replicate', fill in the number of additional rows you require, and click 'OK'. Line List will automatically repeat all of your selections, except for the 'Name' and '@Port' entries, which will be numbered automatically.

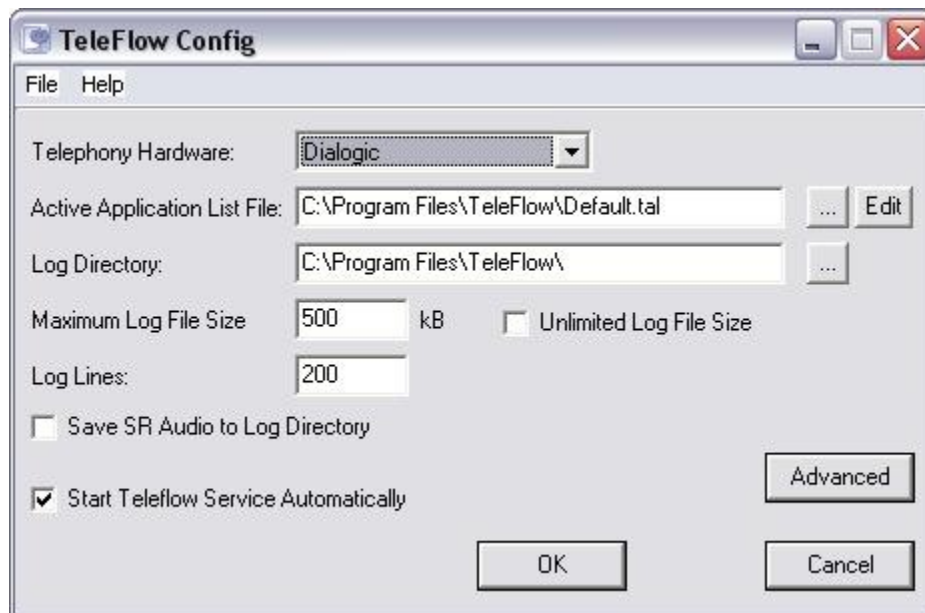
## Application List Fields

Application List Field	Description
Name	A unique name for the line. This will be the name displayed on the <a href="#">TeleFlow Monitor</a> , and the name used for any log file generated for the application instance.
Filename	The TeleFlow Designer created application. (.TAP file)
Protocol	The telephony protocol to use. The options are: fgd0: Feature group D. gds0: Ground start. isd0: ISDN lps0: Analog loop start. mfc0: nocc: ops0: sip0: SIP. wnk0: US Wink Start. (T1 CAS) wnk1:
Restart Count	The number of times the TeleFlow Server should restart a particular application instance after it has terminated. (Typically set to 5 during testing, and 0 when the system is live)
No Limit	If checked, indicates that the TeleFlow Server should restart the application an unlimited number of times, regardless of the Restart Count. This box should be checked in live systems where testing and debugging have been completed.
Log Output	This indicates when the TeleFlow Server should output activity to its log file. It can be set to either 'Errors Only', which means that log output is generated only when a fatal error occurs, or to 'All Events' which means that the TeleFlow Server should constantly output its activity to a log file, regardless of what the activity is. 'Errors Only' is usually adequate for most application debugging. (NOTE: Your operating system may have 'open file limits' that will prevent you from running too many simultaneous applications with 'All Events' on.)

Append to File	Indicates whether or not the TeleFlow Server should append to a log file, or overwrite it. In a live system, this should be checked so that all abnormal application terminations can be monitored.
@Port	The port number to use for this instance of your application. In the example above, the first application instance "inbound01" would use port 1, or the first port on the voice card. The second application instance would use port 2 and so on. The heading of the column is "@PORT" rather than "Port" as a reminder that the @PORT system variable can be used in an application to query the port number the application is running on.
Parameters	Parameters 1 through 9 are created as variables @P1 through @P9, respectively, in your application. These fields allow you to pass values into your application on startup. If your application do not need parameters (many do not), then leave these fields blank.

## 7.3 TeleFlow Config

TeleFlow Config indicates how and what TeleFlow Server should run. See the table below for information on what each of the TeleFlow Config fields is for.





Configuration field	Action/Description
Telephony Hardware	Select the appropriate hardware type from the drop-down list.
Active Application List File	Set this to the .TAL file the TeleFlow Server is to use. TAL files are created with <a href="#">TeleFlow Line List</a> .
Log Directory	This is the directory where log files will be written to. This directory must exist on disk for files to be written to it
Maximum Log File Size (Kb)	This is the largest size, in Kilobytes, that a single log file will reach. NOTE: Every line in the TAL file generates its own log file.
Unlimited Log File Size	Check this box to indicate that there is no upper limit to the size of a log file. This means that log files will continue to grow as long as there is enough space on the hard drive.
Log Lines	This is the number of lines of server output that will be written to log files when an application terminates abnormally. The maximum line count is 20000.
Save SR Audio to Log Directory	Click this check-box to save extra information in you Log Directory for every speech recognition step. If you do not use speech recognition in your application, this setting has no effect.
Start TeleFlow Service Automatically	If you wish to start the <a href="#">TeleFlow Server</a> automatically with Windows (as a Windows service). Consult <a href="#">TeleFlow Server</a> for more information.

## 7.4 TeleFlow Register

TeleFlow Register is used to enter the registration information required to activate a TeleFlow license. Follow the [Licensing Procedure](#) to obtain a TeleFlow license. enGenic will respond with the values required for activation.

Enter all characters in uppercase.

Number of Ports  Transaction ID

MAC Address  Teleflow License   
 (do not include spaces or dashes)

Expiry Date   
 No Expiry Date

Register Cancel

### Registration field

<b>Configuration field</b>	<b>Action/Description</b>
Number of ports	Number of concurrent TeleFlow applications (bound to ports) you are licensed to run.
Transaction ID	The transaction ID, supplied by enGenic, that verifies your purchase of TeleFlow port licenses.
MAC Address	This field uniquely identifies your computer. The value of the field should be filled in automatically. Use the value that is provided unless specifically instructed otherwise by enGenic.
Expiry Date / No Expiry Date	If you have been granted a temporary license, it will have an expiry date. Most of the time you will click the "no expiry date" check-box.
TeleFlow License	License key provided by enGenic

**TeleFlow**

**Part**



## 8 4 - Deploy/Run Applications on TeleFlow Server

**TeleFlow** Server (TFServer.exe) runs **TeleFlow Applications** (.TAP) created in [TeleFlow Designer](#). The .TAP files are run on the telephony hardware (NMS, Dialogic or Ai-Logix). Each channel or port of the telephony hardware runs a .TAP file specified in the .TAL file, which is written in [TeleFlow LineList](#). The TeleFlow Server is responsible for starting the applications, communicating their output to the necessary logs and to the [TeleFlow Monitor](#), re-starting applications that terminate incorrectly, and intercepting any communications sent to it from the [TeleFlow Monitor](#).

Once you have [configured your TeleFlow Server](#), you can refer to the following topics for details on how to run TeleFlow Server:

[TeleFlow Server as a Service](#)  
[Running TeleFlow Server](#)  
[Stopping TeleFlow Server](#)

### 8.1 Running TeleFlow Server as a Service

In order to run the TeleFlow Server service, you must first configure it with TeleFlow LineList and TeleFlow Config. (see chapter 3, "Configure TeleFlow Server")

For production environments, it is usually desirable to have TeleFlow Server run as a Windows service. Special care and consideration must be taken when using TeleFlow Server in this manner because Windows services are different than console applications, and have various restrictions.

#### SERVICE STARTUP

After installation, the Teleflow Server Service (TFServer) is set to manual start-up in the Windows services control manager. Once you have set up your application via [Teleflow Config](#), you can set TFServer to start automatically with Windows. The following precautions must be taken:

##### ODBC drivers

Any ODBC drivers your application might need have to be System DSN's, rather than User DSN's.

##### Location of TAP/TAM files

In your TeleFlow Linelist (.TAL), you cannot access TeleFlow Application (.TAP) files on network drives. All your application files have to be on local drives. Once TFServer is running, and your applications have been loaded and are running, it is still possible to access network resources, but you must use UNC paths to do so.

#### HARDWARE STARTUP

TeleFlow Server will not run as a service unless the hardware required is configured to start with Windows. Each hardware manufacturer handles this differently.

##### Ai-LOGIX

Ai-Logix boards automatically start with Windows. No changes are necessary.

##### Dialogic

Use the Dialogic Configuration Manager (DCM) to configure your boards. Then set the startup mode to be automatic from the DCM Service menu:

Alt + Service -> Startup Mode -> Automatic

### NMS Communications

NMS boards do support starting automatically with Windows but the following changes must be made to your NMS OAM configuration file (usually C:\NMS\OAM\CFG\oamsys.cfg):

- For each board instance, include the following keyword setting:

```
AutoStart=Yes
```

- Include these lines at the end of the file:

```
[supervisor]
AutoStartEnabled=Yes
AutoStopEnabled=Yes
```

- re-run oamsys from the Command Prompt to reload your boards
- NMS boards will now start and stop automatically with the NMS Communications CT Daemon when Windows is restarted.

Example NMS OAM configuration file set to start with Windows:

```
#-----
# oamsys.cfg
#-----
[board0]
Product = AG_2000
Number = 0
Bus = 0
Slot = 9
File = _AG_2000.cfg
AutoStart = Yes

[board1]
Product = AG_QuadDual_T1
Number = 1
Bus = 0
Slot = 13
File = _AG_quaddual.cfg
AutoStart = Yes

[supervisor]
AutoStartEnabled = Yes
AutoStopEnabled = Yes
```

If you have ISDN trunks, you must also install a service wrapper to run each instance of ISDNCTA that you need.

- Obtain the following files from your TeleFlow CD, under \ThirdParty\Utilities\. Copy these files to

your TeleFlow folder (usually C:\program files\teleflow\).

```
instsrv.exe
srvany.exe
srvany.wri
tfdchan.bat
```

- Modify "tfdchan.bat" to use the parameters required by your ISDN installation. If you have more than one trunk, you must create a separate batch file for each trunk. Each batch file must be installed "as a service", according to these instructions.
- Ensure you have Administrative privileges on the PC to install the services.
- Install the SRVANY service as follows (refer to 'srvany.wri' for more information):

```
INSTSRV "TFDChan01" "c:\program files\teleflow\srvany.exe"
```

Run the Windows Registry Editor (regedit.exe), and make the following changes, under HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\TFDChan01:

- create a string value called "DependOnService" and set that value to "ctdaemon".
- create a "Parameters" key.
- under the "Parameters" key, create a string value called "Application", and set that value to "TFDChan.bat".
- under the "Parameters" key, create a string value called "AppDirectory", and set that value to "C:\NMS\bin".
- Use the Windows Service Control Manager (Start -> Settings -> Control Panel -> Administrative Tools -> Services) and find the "TFDChan01" service. Change the startup type to "Automatic".

## 8.2 Running TeleFlow Server in Console Mode

In order to run the TeleFlow Server service, you must first configure it with TeleFlow LineList and TeleFlow Config. (see chapter 3, "Configure TeleFlow Server")

You must also have your telephony hardware and drivers installed. The drivers must be configured appropriately, and any services the telephony hardware depends on must be running. Each hardware manufacturer has different installation and configuration procedures for their products. For more information, consult the [Hardware Installation](#) appendix.

### Run TeleFlow Server

You can start TeleFlow Server from the console in one the two following ways:

- (1) From a Windows Command Prompt:
  - Click "Start" --> "Programs" --> "Accessories" --> "Command Prompt"
  - Type "TFServer" and press Enter
- (2) From the Windows "Run" dialog:
  - Start --> Run
  - Type "TFServer"

- Click "OK"

Whether you use (1) or (2), You should see the message "TFServer running in console mode." in a Command Prompt window, similar to this example:

A screenshot of a Windows Command Prompt window. The title bar reads "Command Prompt - TFServer". The command prompt shows the command "C:\>TFServer" and the output "TFServer running in console mode." The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

### TeleFlow Server as a Service

For production environments, it is usually desirable to run TeleFlow Server as a Windows service. If this is how you wish to start TeleFlow Server, consult [TeleFlow as a Service](#) for more information.

## 8.3 Stopping TeleFlow Server

See the [Safe shutdown for TeleFlow Server](#) topic for details on shutting down TeleFlow Server.

**TeleFlow**

**Part**





## 9 5 - Monitor and Maintain TeleFlow Applications

Once your TeleFlow Server is running, you will want to monitor its success, and ensure everything is running as expected.

The following chapters discuss some of the tools and utilities available for TeleFlow Server maintenance, and how to perform basic maintenance procedures:

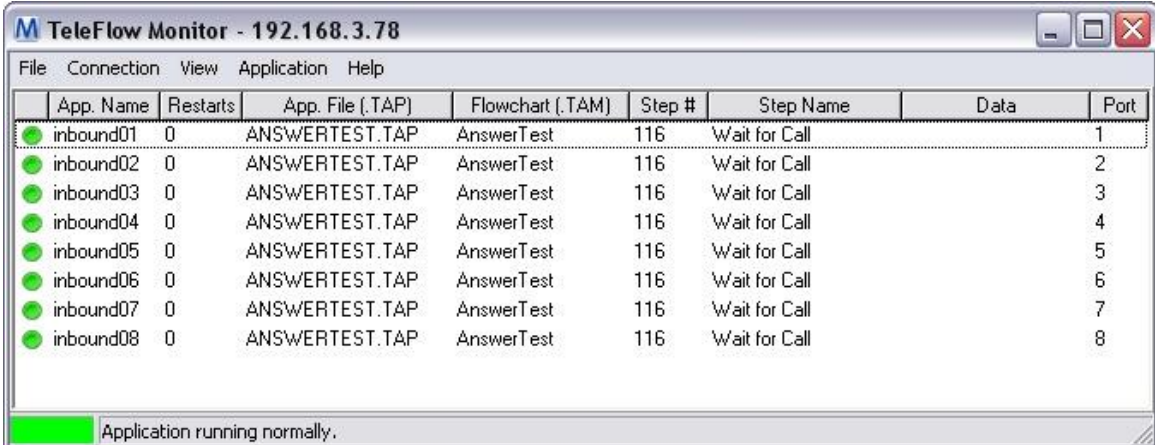
[TeleFlow Monitor](#): This 'status monitor' displays the activity occurring on all TeleFlow applications currently running on the specified Voice Application Server.

[System Health](#): The SystemHealth web page provides an up to date overview of the system as it stands.

[Safe shutdown](#): Information on how to safely shutdown a TeleFlow Server.

### 9.1 TeleFlow Monitor

The TFMonitor is the visual monitor of TeleFlow applications as they run. The picture below shows a TFMonitor that is displaying the status of 8 applications instances running on a remote IVR computer.



The screenshot shows a window titled "TeleFlow Monitor - 192.168.3.78". The window contains a table with the following data:

App. Name	Restarts	App. File (.TAP)	Flowchart (.TAM)	Step #	Step Name	Data	Port
inbound01	0	ANSWERTEST.TAP	AnswerTest	116	Wait for Call		1
inbound02	0	ANSWERTEST.TAP	AnswerTest	116	Wait for Call		2
inbound03	0	ANSWERTEST.TAP	AnswerTest	116	Wait for Call		3
inbound04	0	ANSWERTEST.TAP	AnswerTest	116	Wait for Call		4
inbound05	0	ANSWERTEST.TAP	AnswerTest	116	Wait for Call		5
inbound06	0	ANSWERTEST.TAP	AnswerTest	116	Wait for Call		6
inbound07	0	ANSWERTEST.TAP	AnswerTest	116	Wait for Call		7
inbound08	0	ANSWERTEST.TAP	AnswerTest	116	Wait for Call		8

At the bottom of the window, a green bar indicates "Application running normally."

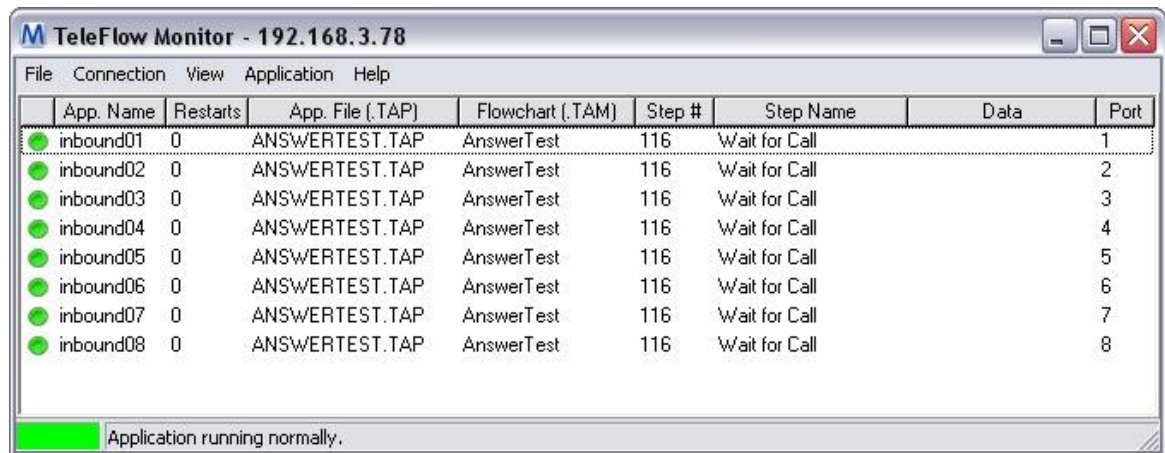
This 'status monitor' displays the activity occurring on all TeleFlow applications currently running on the specified IVR computer. Activity is continuously monitored, then reflected onto the screen. The monitor indicates the status of each application instance through the use of 'status lights'. The status lights occupy the very first column in the Monitor. The possible displays of the status lights are explained below under "Using TFMonitor". The rest of the columns are as follows:

TeleFlow Monitor Column	Description
App.	A unique name for each application instance. Defined in <a href="#">TeleFlow Line List</a> as "Name"
Restarts	Keeps track of how many times this instance of an application has terminated and been restarted by TeleFlow Server.
App. File (.TAP)	The TeleFlow Designer created application. (.TAP file)

Flowchart (.TAM)	The name of the flowchart or TeleFlow Application Module (.TAM), created using TeleFlow Designer, which is currently being run by TeleFlow Server.
Step #	The last executed Step number within the current TeleFlow Application Module (.Tam).
Step Name	The descriptive name of the type of TeleFlow Step the application is executing.
Data	Optional data that can be passed from your application to the TeleFlow Monitor by using the System Variable <a href="#">@SYS_DATA</a> .
Port	If the application is running on telephony hardware, and there is an active call, this is the current port or channel that a call is taking place on.

## Starting TFMonitor

TeleFlow Monitor uses TCP/IP to connect to and monitor a running [TeleFlow Server](#) program. The first time TeleFlow Monitor runs, it will start blank and a connection must be chosen. In the menu list select Connection and then connect. The connection dialog will appear:



In the "Host:" field, supply a host name or IP address to connect to. The default is "localhost", which is the current computer. The "automatically connect" checkbox allows you to instruct the Monitor to automatically attempt this connection when it starts.

When the Monitor establishes a connected, the box in the lower right will turn green, as in the picture above.

## Using TFMonitor

The TFMonitor displays the active step of each application instance. Left click on one of the lines and choose the menu option - Application (right clicking will bring up this menu as well). The Application menu will give you the options to Reload Shutdown the application instance.

You also have the option to "Show Messages". This will open a window and display the current log output of that application instance up to the most recently executed Step. The number of lines displayed is dependent on your "Log Lines" setting in [TeleFlow Config](#). You can display an application's log in this fashion even if that application's "Log output" setting in [TeleFlow Line List](#) is set to "Errors only".

If a **red** light appears to the left of the App. Name, the application had a fatal error. A log entry is made after a fatal error, and the location of this file is set in [TeleFlow Config](#). The application will either restart or stop at the fail depending on the setting "Restart Count" in [TeleFlow Line List](#).

If a **yellow** light appears to the left of the App. Name, the application restarted after a fatal error and a log entry has been made.

A **bright green** light to the left of the App. Name indicates that the application is running normally and has not suffered any terminations since the last time it was started, either by start [TeleFlow Server](#) for the first time, or by using the Reload option described above.

A **dark green** light indicates that the application has terminated normally. This occurs when an application is shutdown manually from the TeleFlow Monitor, or the application reached its ending point (as programmed).

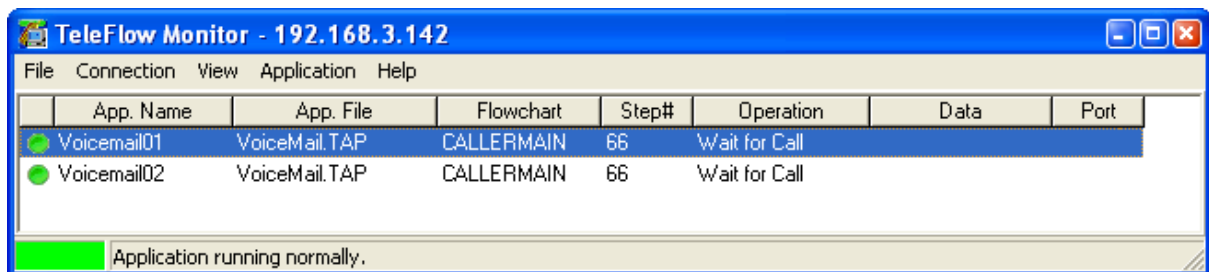
A **bright green** light to the left of the App. Name indicates that the application is running normally and has not suffered any terminations since the last time it was started, either by start [TeleFlow Server](#) for the first time, or by using the Reload option described above.

A **dark green** light indicates that the application has terminated normally. This occurs when an application is shutdown manually from the TeleFlow Monitor, or the application reached its ending point (as programmed).

Use this link for more information on the [TFMonitor menus](#).

### 9.1.1 TeleFlow Monitor Menus

The various menu options of the TeleFlow Monitor are explained below.



#### FILE MENU

##### Options:

- Opens the TeleFlow Monitor options window.

##### Reload Application List:

- Reloads the current TeleFlow Application List(.TAL) file, including any changes that have

been made to it since the last time it was loaded.

- Applications to reload cannot be active. (See Application for shut-down procedures)
- The .TAL file to load is determined by TFConfig.

**Exit:**

- Exit the TeleFlow Monitor

**CONNECTION MENU****Connect:**

- Connect to a TeleFlow Server. (Start Monitoring)
- If the monitor is run on the same computer as the TeleFlow Server it is to monitor, choose localhost when the connection dialog appears. If it is to monitor a Server across a network, enter the IP address of the computer the Server is running on.
- **IMPORTANT:** The TCP/IP protocol must be installed and configured on both systems for the monitor and Server to communicate at all.

**Disconnect:**

- Disconnect the TeleFlow Server connection. (Stop Monitoring)

**VIEW MENU****Refresh:**

- Refreshes the data in the Monitor window immediately.
- Normally, the information displayed by Monitor is periodically refreshed with information from the TeleFlow Server. The frequency of these refreshes can be set in the Options window. (See File Menu)

**APPLICATION MENU****Show Messages:**

- Shows TeleFlow Server output - or the activity that has occurred up to the time Show Messages was clicked - on the application currently selected in the monitor console.

**Reload Normal:**

- Reloads the selected application **ONLY** when the application is set to [RELOAD ON](#).
- Reloads a selected application that was previously shut down.
- This will refresh the .TAM and .TAP files if updated versions have been put in place.

**Reload Abort:**

- Shuts down & reloads the selected application immediately.
- Reloads a selected application that was previously shut down.
- This will refresh the .TAM & .TAP files if updated versions have been put in place.

**Shutdown Normal:**

- Shuts down the selected application **ONLY** when the application is set to [RELOAD ON](#).

**Shutdown Abort:**

- Shuts down the selected application immediately.

**Reload All Normal:**

- Reloads every application ONLY when the application is set to [RELOAD ON](#).
- Reloads applications that were previously shut down.
- This will refresh the .TAM and .TAP files if updated versions have been put in place.

**Reload All Abort:**

- Shuts down & reloads ALL applications immediately.
- Reloads applications that were previously shut down.
- This will refresh the .TAM & .TAP files if updated versions have been put in place.

**Shutdown All Normal:**

- Shuts down every application ONLY when the application is set to [RELOAD ON](#).

**Shutdown All Abort:**

- Shuts down ALL applications immediately.

**HELP MENU****About:**

- Opens the TeleFlow Monitor's 'Help About' box.

## 9.2 SystemHealth


The SystemHealth web page provides an up to date overview of the system as it stands. A variety of information is provided about items related to TeleFlow software and the other components that compress the Voice Applications Server.


To see the system health page on a system with TFNetHub installed, type the address into the a web browser.

For example:

<http://localhost:78/systemhealth.html>

<http://192.168.3.11/systemhealth.html>





## System Health

---

<b>System Name</b>	<b>TFVO Office System</b>
<b>Report Created</b>	2006/06/12 ( <b>Monday</b> ) 08:58:26
<b>NetHub Started</b>	2006/05/30 ( <b>Friday</b> ) 08:53:42
<input type="checkbox"/> Run Time	263 Hours
<input type="checkbox"/> Version	1.2.1.48 (Service)
<input type="checkbox"/> Data Served	75 MB
<input type="checkbox"/> Data Received	7 MB
<b>Notes</b>	PBX Conferencing System and IVR
<b>Computer Name</b>	UMBRIEL
<input type="checkbox"/> Operating System	Windows 2003 (Build 3790: Service Pack 1)
<input type="checkbox"/> CPU	3.0 GHz
<input type="checkbox"/> # Processors	2 (1 if Hyperthreaded)
<input type="checkbox"/> RAM: Physical	1051 MB Free of 2047 MB Total
<input type="checkbox"/> RAM: Virtual	2016 MB Free of 2048 MB Total
<input type="checkbox"/> IP Address	192.168.3.56
<input type="checkbox"/> Drive C: = 20 GB Free	<div style="width: 100%; height: 15px; background-color: #e0e0e0; position: relative;"><div style="width: 20%; height: 100%; background-color: #008000;"></div></div>
<b>System Reports</b>	<input type="checkbox"/> None found
<b>Service Display Names</b>	
<input type="checkbox"/> TeleFlow Server	<div style="background-color: #008000; color: white; padding: 2px;">RUNNING (TFServer)</div>
<input type="checkbox"/> TeleFlow Server Time Service	<div style="background-color: #008000; color: white; padding: 2px;">RUNNING (TFServerTimeService)</div>
<input type="checkbox"/> MySQL	<div style="background-color: #008000; color: white; padding: 2px;">RUNNING (MySQL)</div>
<input type="checkbox"/> Apache	<div style="background-color: #ffcc00; padding: 2px;">NOT INSTALLED</div>
<b>TeleFlow Licensed</b>	Forever
<input type="checkbox"/> For Ports	200
<input type="checkbox"/> Server Started	2006/06/02 ( <b>Friday</b> ) 09:39:45
<input type="checkbox"/> Run Time	271 Hours
<input type="checkbox"/> Version	Build 2006.0506
<input type="checkbox"/> Last Started	2006/06/09 11:15 (3 days)
<input type="checkbox"/> Installed Dir	C:\Program Files\TeleFlow\
<input type="checkbox"/> Log Dir	C:\TFVO\log\
<input type="checkbox"/> Log Lines/Max	1000 / Unlimited
<input type="checkbox"/> Log Size	27 MB
<b>TeleFlow Monitor</b>	
<input type="checkbox"/> Normal	<div style="display: inline-block; width: 10px; height: 10px; background-color: #008000; border-radius: 50%;"></div> 44
<input type="checkbox"/> Stopped	<div style="display: inline-block; width: 10px; height: 10px; background-color: #ff0000; border-radius: 50%;"></div> 0
<input type="checkbox"/> Restarted	<div style="display: inline-block; width: 10px; height: 10px; background-color: #ffff00; border-radius: 50%;"></div> 0
<input type="checkbox"/> Failed	<div style="display: inline-block; width: 10px; height: 10px; background-color: #000000; border-radius: 50%;"></div> 0

### System Name

The system name describes the system as it is referred to by the systems administrators. This name can be changed in the TFNetHubEditor program.

### Report Date and Time

This is the time that the page was refreshed last. The system is set to refresh the page every 10 minutes; however F5 can be pressed on the webpage to get an immediate update.

### Computer Name and System Information

This section provides information about the computer hardware and operating system. Please note that hyper-threaded CPUs will register as two processors.

### TeleFlow Licensed

Information about TeleFlow Server software is provided in this section. Version information, run time and license status is provided. Information about the size of the log directory is also available. This is related to the TFLineList Errors section below as well as Speech Recognition. Because there is a fairly intensive process required to read the size of the log directory, the process is only done occasionally. Consequently, the Log Size information may be a couple hours old.

### System Reports

This section contains information that other processes have provided to SystemHealth. For example, data transfer programs and processes will write information about successes or failures. Should the last transfer happen too far in the past, there may be an indication of a transfer issue. Accordingly, the box will display orange if there might be a problem and red if there certainly is.

### Services

Installed services related to TeleFlow applications are displayed here. Should a service process show as "Stopped", a red box will indicate that the process may have shutdown and not restarted. A stopped service probably means that the system, or part of it, is no longer operational.

### TFLineList Errors

"Line Restarts" will occur from time to time on a brand new system as the bugs are worked out. However, once the system is in production, these should occur rarely if ever. Should a line restart occur, it probably means a caller was dropped. When this happens, the log file that captured the event will show as red for the first hour and then orange for the next 72 hours. Should there be a significant number of red bars, it may mean the system has failed operating. For example, a possible reason why this may occur could be due to a corrupt database.

## 9.2.1 Configuration of SystemHealth

### TFNetHub

The SystemHealth.html web page is created by TFNetHub. To change information pertaining to SystemHealth, start the settings with TFNetHubEditor. Please note that TFNetHub will need to be restarted in order for the changes to take effect.

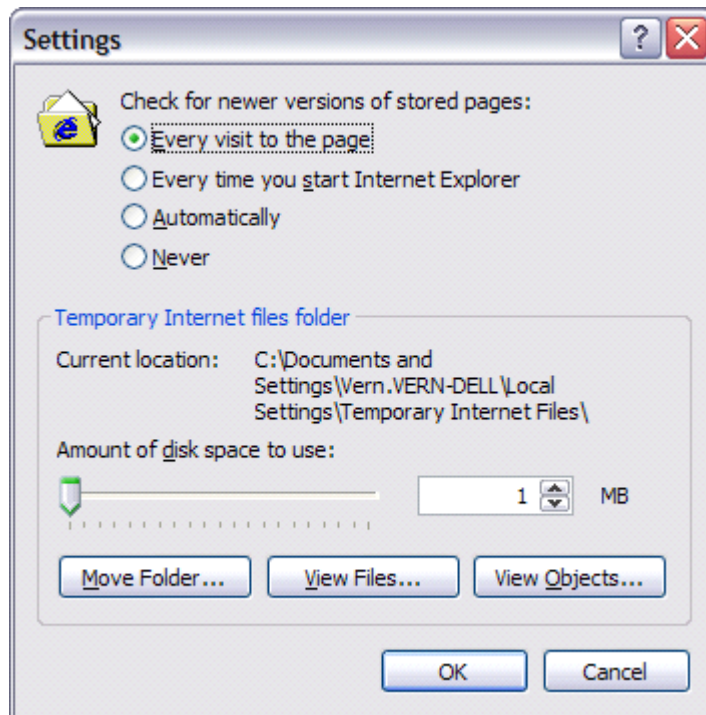
“Refresh Time” should be set to approximately 10 minutes, but can be set to any number equal to or greater than a minute. This number should not be too frequent as building the systemhealth.html webpage can be processor intensive.

### Web client

On the client end web browser, a zero cache should be set in order to show an error page should TFNetHub become unreachable. This may in turn in itself indicate an error on the Voice Application server.

The following setting shows the correct setting for Internet Explorer.

Tools > Internet Options > General > Settings





## 9.3 Safe shutdown for TeleFlow Server

### Shutdown and restart

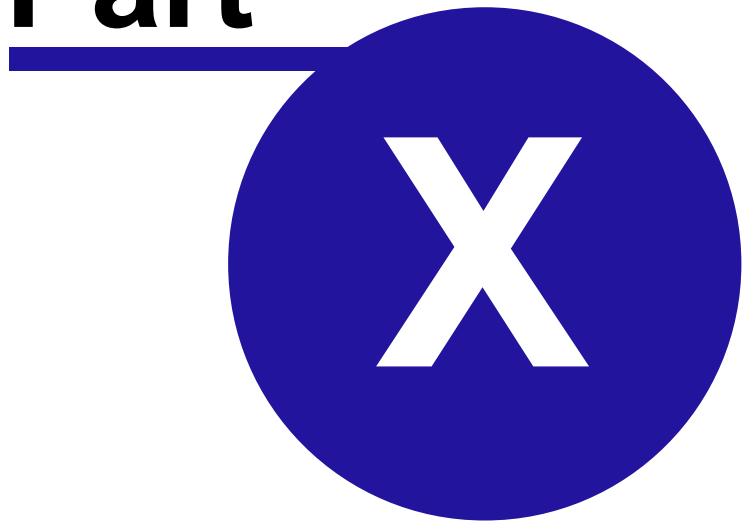
Unless the system has completely locked up, and an orderly shutdown is not possible, a Voice Application Server should be shut down in the following manner:

1. On the [TeleFlow Monitor](#), click 'Application' and then 'Shutdown All Normal' (If the [TeleFlow Monitor](#) is not maximized, you should be able to locate it in the Windows taskbar. If it isn't running, find it in the TeleFlow Windows Programs menu, under "Server Tools"). For additional information about TeleFlow Monitor functionality, refer to the chapter on [TeleFlow Monitor](#).
2. Wait for all of the light green indicator lights in the left-hand side of the window to turn dark green. NOTE: The lights will remain light green for one of the following reasons:
  - a. The applications haven't completed the tasks necessary for a 'safe shutdown'. Such conditions might be that a caller is currently on the line, or a report that takes awhile to complete.
  - b. The required logic wasn't implemented in one of the applications. Generally, if all callers are finished, and you have given the system a few minutes to clear the remaining lights, but there are some light green lights, you should probably continue with step 3 as though all lights are dark green.
3. Once the indicator lights have all turned dark green, it is safe to shut down the remaining applications. Be sure to shut down any applications you are running. Most software required for running TeleFlow Applications will safely shut down with Windows.
4. Click 'Start', 'Shut Down' and select 'Restart the Computer', then click the 'Yes' button. DO NOT use the option to 'Close all programs and log on as a different user'.

If some or all of the indicator lights remain green, and an immediate shutdown is required, you can start the shutdown process as per point 3, above. Be aware, however, that if a caller is on a line, or an important report is being run, both of these operations will be stopped abruptly. This may result in reports not being completed when the system is restarted, or unexpected data in TeleFlow Application(s) data files.

**TeleFlow**

**Part**



## 10 Reports and statistics

TeleFlow Stats (TFStats.exe) is an easy way to track call statistics in a TeleFlow application. TeleFlow Designer has the template tam Logger. Adding this into your application and entering text into the first parameter field will show up in a log file that TFStats.exe can read.

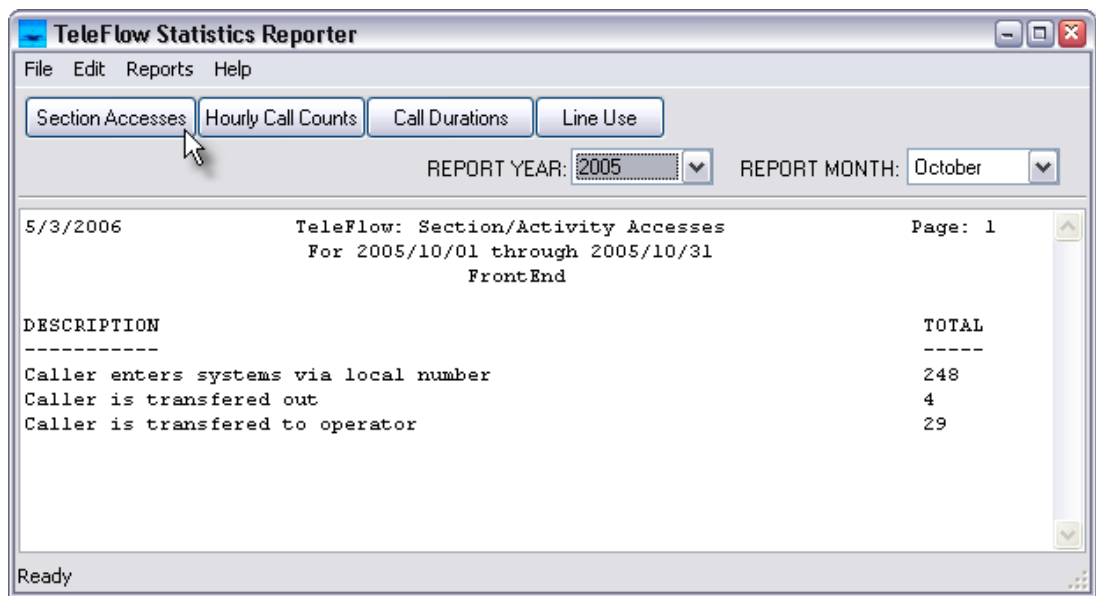
Quick Links

[How to use TeleFlow Statistics Reporter](#)

[TeleFlow SQL Reporter](#)

[How to Create an SQL Reporter file](#)

### 10.1 How to use TeleFlow Statistics Reporter



- Open TFStats.exe
- Choose Report Year; e.g. 2005
- Choose Report Month; e.g. October
- Four choices of reports
  - Section Accesses – States the number of times a given event occurred over the course of the month.  
Example: 'Caller chooses to use touch tone mode' with a total of 3965 means that 3965 callers selected Touch Tone mode rather than Speech Recognition mode.  
See the 'Data Collected' section below for information about events collected in your Voice Application Server(s).
  - Hourly Call Counts – number of calls by day and by hour
  - Call Durations – minutes of access by day and by hour
  - Line Use – number of FrontEnd access entries by port

## Section Accesses

The Section Accesses is an overall call count and accesses for all ports. Please check appendix A for specific client listings.

The LOGGER.Tam file is writing to text based log files in the \ClientDir\Log directory with the following naming convention:

\ClientDir\Log\TF20010100.Log

- The 'TF' stands for TeleFlow.
- '2001' represents the current year.
- '01' represents the current month. I.e.: January
- '00' is used if all logging for the month will go in the same file. If logging is daily, then '01' ...'31' will be used in place of '00'.

The entries in the log file are comprised as follows:

2000/03/22,20:00:32,2,361,HANGUP,71111,921,BIGSYS,BC,VAN,CR,ELL0236E,1132,X

The first four fields will always remain as they are now, and represent the header stub of the log entries.

DATE: '2001/01/22' is the date of the log entry.

TIME: '20:00:32' is the time of day (24 hour format) the entry occurred

LINE: '2' represents the telephone line that handed the group

VERSIO '361' is the version number for logging. This is used to determine the structure of the following fields.  
N:

**Important:** If Logging needs to change from the 361 standard, then a new version number must be put in place, and defined in this document, so that stats and related applications can know how to deal with the new data.

### VERSION 361

The following fields are written for the logging file.

5	SinceMid	'71111'	is the number of seconds since midnight.
6		'921'	(@CALL_START)
7	Elapsed	'	is the elapsed seconds since the call started.
8	SysName	BIGSY	(@CALL_ELAPSE)
9		S'	is the name of the system / system code.
10	SystemCode	'BC'	is the location of the system (Could be physical or by
11		'VAN'	organization code)
13	OfficeLocation	'CR'	is the office location (Generally city, or specific office
14	ClientType	'	code)
		ELL023	is the client type (NA for not available)
	ClientID	6E'	is the client account number or identification code
	ClientPIN	'1132'	is the client pin or security password
	Data	'X'	is additional data from the TeleFlow

### Save Options

After running any of the reports, you can choose to save it. By choosing "Save As" in File menu, you will have options as to where to save them, and in what format.

If you save the file as a Text file, it will be formatted just as it is in the TFStats output window.

If you save it as a CSV file, only the raw data that was used in the formatted report will be saved, in CSV format. This allows it to be easily imported into other data manipulation tools for further processing.

The following section describes the contents of each of the CSV files that are created when doing a "Save As" to CSV file for each of the report types.

#### Section Accesses:

Saves a single file with the following "fields".

Field #	Description
1	Call activity log event code
2	Call activity log event code report description
3	Number of times the event occurred

#### Hourly Call Counts:

Saves three files. The first has the name you supplied, and gives the count of calls for each hour for each day. The second has the name you supplied, with '\_hourly\_tot' added to the name, and gives the total calls for each hour of the day for the entire month. The third has the name you supplied, with '\_daily\_tot' added to the name, and gives the total calls for each day for the entire month.

File # 1

Field #	Description
1	Date of the count, in YYYY/MM/DD format
2	Hour of the count, for the date in field#1 (Eg. For midnight to just prior to 1 am, this field contains 00:00:00). In HH:MM:SS format, although the minutes/seconds will always be :00:00
3	Number of calls for the hour

## File # 2

Field #	Description
1	Hour of the count for the month (Eg. For midnight to just prior to 1 am, this field contains 00:00:00). In HH:MM:SS format, although the minutes/seconds will always be :00:00
2	Number of calls for the hour of the day, for the entire month

## File # 3

Field #	Description
1	Day of the count for the month, in YYYY/MM/DD format
2	Total number of calls for the day

Call Durations:

Saves three files. The first has the name you supplied, and gives the total minutes of a calls for each hour for each day. The second has the name you supplied, with '\_hourly\_tot' added to the name, and gives the total minutes for all calls for each hour of the day for the entire month. The third has the name you supplied, with '\_daily\_tot' added to the name, and gives the total minutes for all calls for each day for the entire month.

## File # 1

Field #	Description
1	Date of the count, in YYYY/MM/DD format
2	Hour of the count, for the date in field#1 (Eg. For midnight to just prior to 1 am, this field contains 00:00:00). In HH:MM:SS format, although the minutes/seconds will always be :00:00
3	Number of minutes of calls for the hour

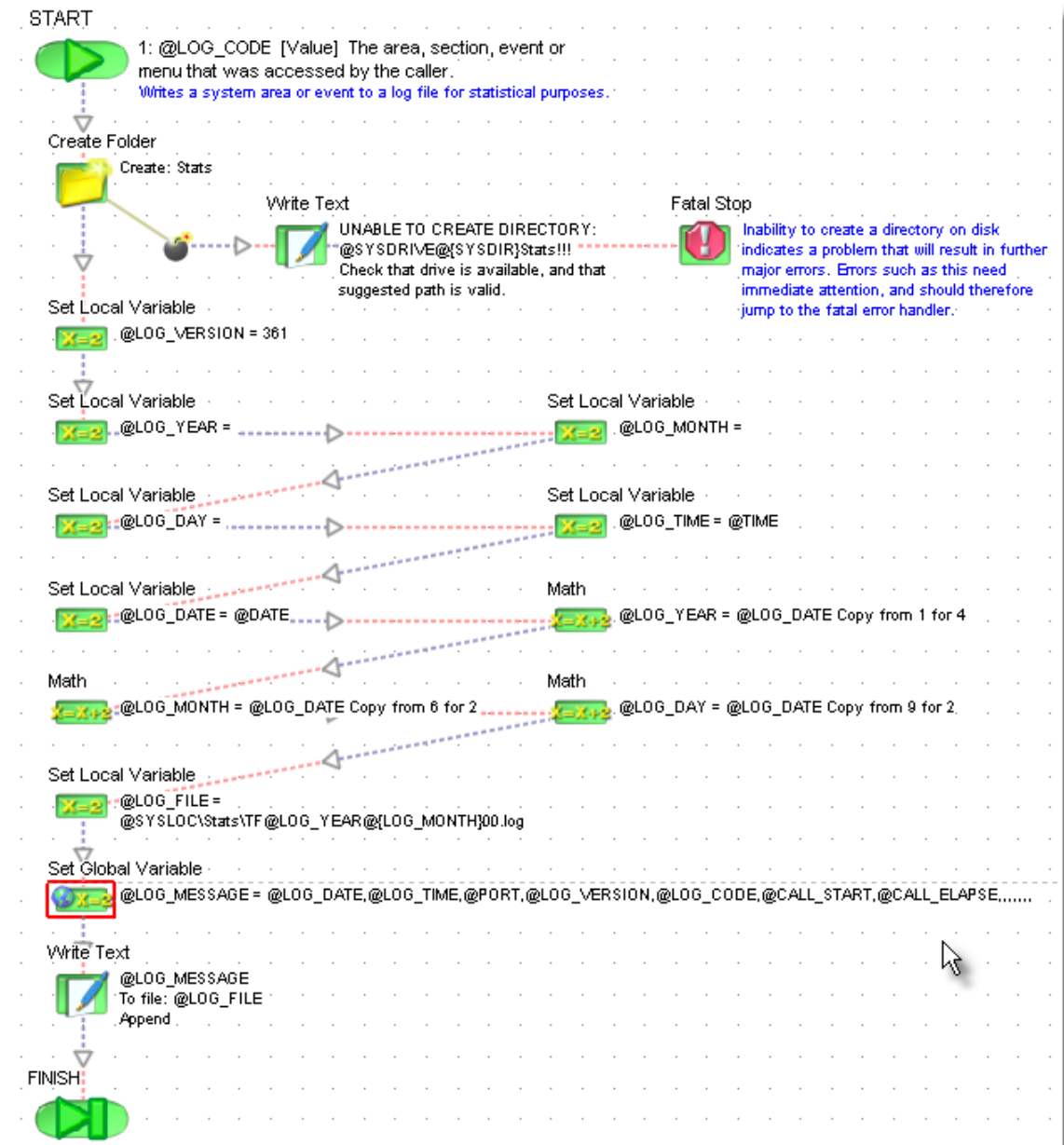
## File # 2

Field #	Description
1	Hour of the count for the month (Eg. For midnight to just prior to 1 am, this field contains 00:00:00). In HH:MM:SS format, although the minutes/seconds will always be :00:00
2	Number of minutes of calls for the hour of the day, for the entire month

## File # 3

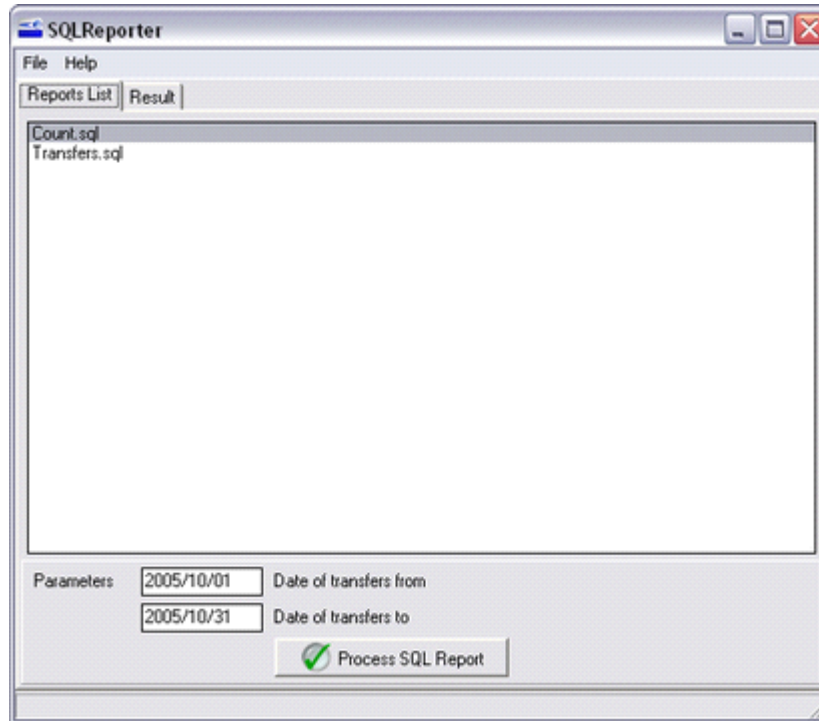
Field #	Description
1	Day of the count for the month, in YYYY/MM/DD format
2	Total number of minutes of calls for the day

This version generally has TeleFlow code that looks like this.



## 10.2 TeleFlow SQL Reporter

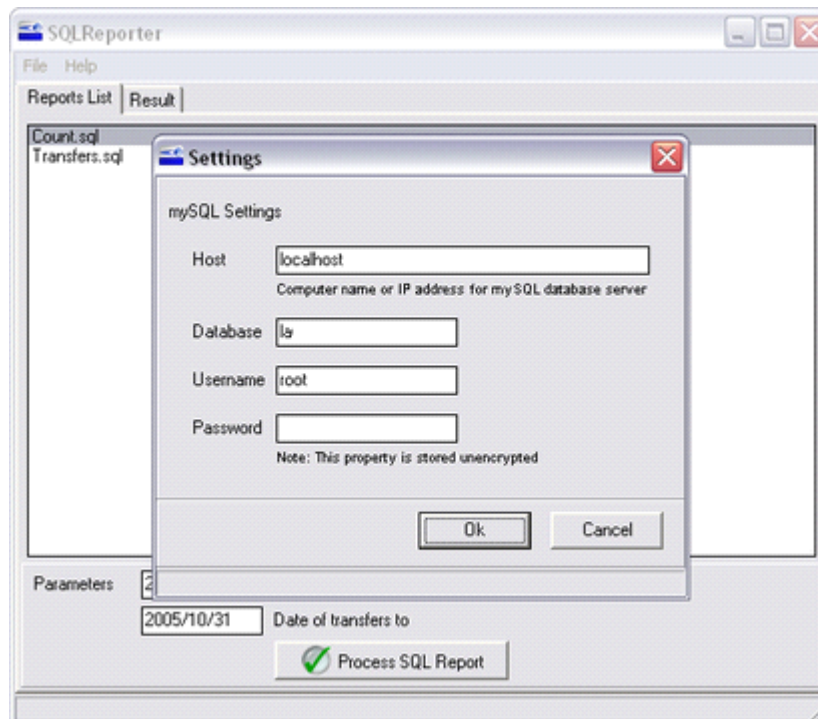
### How to Use



- Choose which SQL (.sql) file to be reported on from the list of available reports (Reports List tab).
- Choose the date range for the report in YYYY/MM/DD  
Note: These parameters may change for each report.
- Click on the button **Process SQL Report**
  - If you wish to save the output click on **Save data to CSV file**

### How to change database





- Click on **File**
- Click on **SQL Settings**
- Choose the
  - host
  - database
  - username
  - password
- Click **OK**
- Close and reopen SQLReporter.

### 10.3 How to create an SQLReporter file

The following example is a file with an extension of .sql. The – define information about input and output. The first none – line needs to be the word “select”. The full SQL statement follows.

```

//////////Example Start//////////
--Title: Phone Numbers
--Title: Transfers made from %P1% to %P2%
--Title: - Printed at %DATE% %TIME%
--Title:
--P1: 2005/10/01    -- Date of transfers from
--P2: 2005/10/31  -- Date of transfers to
--F: phone_number  S 15  . -- Phone Num
--F: BIGC          S 15  . -- count
--End:
--End: Total of %ROWS% reported
--End:
select
phone_number,
count(phone_number) as BIGC,
transfer_date
from

```

```

    transfers
    group by
    phone_number
    having
    transfer_date >= '%P1%'
    and transfer_date <= '%P2%'
    order by
    BIGC desc
//////////Example End//////////

```

The sql files are formatted to the following specifications:

**--Title: Phone Numbers**

This is a title line in the report.

You may include parameters (%P1%), date (%DATE%) or time (%TIME%)

%P1% ... %P9% are parameters passed to SQLReporter from the GUI.

**--P1: 2005/10/01 --Date of transfers to**

This is a parameter field where %P1% can be used in the sql statement

**--End: Total of %ROWS% reported**

An end statements (--End:) will show in the end of the report. This line will report the total number of rows (using %ROWS%) retrieved from the sql statement.

```

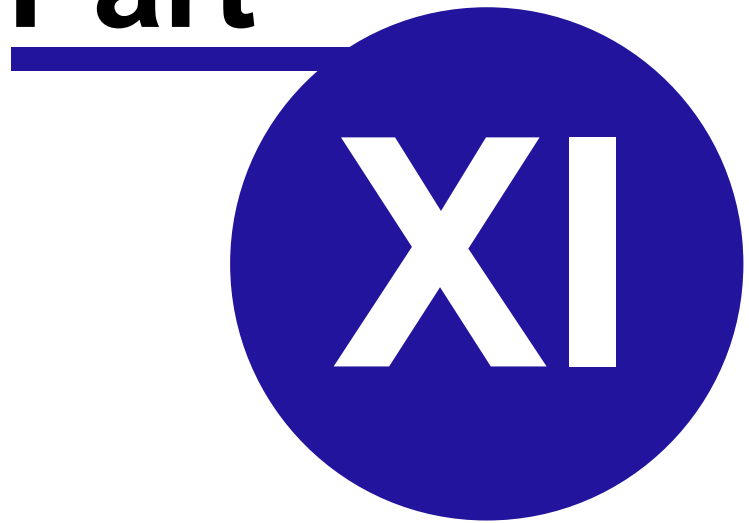
select
    phone_number,
    count(phone_number) as BIGC,
    transfer_date
from
    transfers
group by
    phone_number
having
    transfer_date >= '%P1%'
    and transfer_date <= '%P2%'
order by
    BIGC desc

```

Follow with the sql statement that is to be run against the database.

**TeleFlow**

**Part**



## 11 TeleFlow FAQ's

### GENERAL FAQ's

#### *Bridge a Call*

**Question:**

I am wondering if there is a way to bridge a call so we could record the conversation after transferring a call.

**Answer:**

Yes you can. Using a transfer done with switching - TeleFlow ships with Example apps that do this (TransferDemo and TransferDemo2) - you can 'remain on the line' after transferring the call. This means that you can also start a recording at any time after that call is started.

If the "trigger" to start recording is something other than the beginning of the call, such as a user having to key in a code to start, you would need to make modifications to the TransferDemo application you are using to react to that touch tone sequence. We recommend that you use a sequence of touch tones, and not just a single tone. Otherwise, the recording might get started unexpectedly with an accidental key press. Otherwise, you can simply follow the Place Call's success path with a Record step to capture the conversation.

#### Chat Line

**Question:**

Can TeleFlow support functions required in developing a real time chat line?

**Answer:**

Yes. Several chat lines have been built with TeleFlow using our switching and conferencing abilities on NMS.

#### Deploy TeleFlow Application

**Question:**

How do you deploy TeleFlow applications? The designer provides a .tap file where as the TeleFlow config requires a .tal file for the active application.

**Answer:**

First, on the production system, you need to have all of the files that make up the TeleFlow application on disk. (TAP files, Tam directory, script and prompt files, along with any specific file requirements of your application, such as database files)

You must define a TAL file using [TeleFlow LineList](#)

You must then set the TAL file active in [TeleFlow Config](#).

The TAL file instructs the TeleFlow Server what to run on which ports, how many ports you have, etc. The TeleFlow help file has a section on building an application list (See: TeleFlow Server >> Server Tools >> TeleFlow Line List)

#### Dictation Step

**Question:**

Is it possible to recover a telephony audio file (pcm 8Khz 16b mono) if it has been cut off in middle of a dictation? (Server restarted and cut off the dictation) I would like to know if it is

possible to recover the audio that was made before the cut off.  
The file is 3,317 KB but does not play.

**Answer:**

First, make sure you work with a copy of this file.  
Use a binary editor on the file, and remove the first 44 bytes, then save it.  
Open the file in CoolEdit as RAW, PCM 16-bit 8KHz Mono. Save the file as a Windows PCM file. Assuming it is possible to recover it, it should be recovered at that point.

## E1 PBX Integration with AI-Logix

**Question:**

For E1, which is the best protocol to assure best integration: CAS or ISDN? Could we use TeleFlow for inbound and outbound call with one E1? Could I transfer a call without use bridge on TeleFlow side?

**Answer:**

ISDN is superior to CAS, so I would use it wherever possible.

You can use TeleFlow for inbound and outbound calls with one E1. For best results, you should configure your in-bound hunting (calls from your switch to your Ai-Logix card) to be **least idle**, top down. On the TeleFlow side, setup your application so that it selects out-bound channels, least idle, bottom-up.

You cannot transfer a call without using a bridge on the TeleFlow side with that Ai-Logix card. You would have to use a feature of your switch called "release link capability", sometimes known as "2B transfer". This feature goes by different names depending upon the switch manufacturer, and what part of the world you are in. Release-link is only available on ISDN, and only certain protocols. It is important to check with the card manufacturer to ensure that the protocol you want to use to get release-link is also supported on the card you want to use.

On E1 circuits, that card supports the following formats:

- Australia Telecom (Austel ISDN TS013-1990)
- ETSI Euro ISDN (ETS 300 102)
- QSIG

## Error Notification

**Question:**

Is there a way to have TeleFlow send an email notification out if it goes down?

**Answer:**

Depending on the nature of what happened when TeleFlow stopped running, you may be able to trigger certain events when it stops. (Was it simply not running, or was the system crashed/frozen?) Assuming you are running TeleFlow Server from the command line (not as a service), you can create a batch file with the following content. You can run whatever program you want after the line that says "rem \* If TF\_Server exits, the following code will run". (So, you could set up a command-line email product to send a particular message to a particular email address in the event of the server stopping.)

Of course, if this batch file works properly, TeleFlow Server would be automatically restarted in the event of a failure. Note that the section in which you run added code after the server stops would also run if you Ctrl-C the batch file, so we would recommend simply using the 'X' to close the window when you wish to stop the server intentionally.

In addition, anything at all you can tell us about the state of the system when the server stopped might help us in determining what went wrong, since we would much prefer to solve the underlying problem...

#### BATCH FILE CONTENTS:

```
@echo off
rem TF_ServerLoop
rem -----
rem Starts TeleFlow Server in a infinite loop to automatically
rem recover from errors.
rem (c) 2000, 2001 enGenic.
rem All rights reserved.
rem
rem Parameters:
rem -----
rem %1 Drive where the application is installed D:
rem %2 System location CanMtg
rem
rem slashes will be added automatically, e.g.:
rem %1\%2\bin
rem

mode 80,100
title TF Server Loop

echo _____
echo.
echo TeleFlow Infinite Server Recovery System
echo (c) 2000 Engage! Software Corp. All rights reserved.
echo _____
echo.
echo Now loading...

rem * Capture server startup date and time to log
date/T >> %1\%2\bin\TF_ServerLoop.log
time/T >> %1\%2\bin\TF_ServerLoop.log
echo >> TFServer Loop Started %1\%2\bin\TF_ServerLoop.log
echo ----- >> %1\%2\bin\TF_ServerLoop.log

:LOOP

rem * Capture date and time of crash to tracking file
date/T >> %1\%2\bin\TF_ServerLoop.log
time/T >> %1\%2\bin\TF_ServerLoop.log
echo >> TFServer start of loop %1\%2\bin\TF_ServerLoop.log
echo ----- >> %1\%2\bin\TF_ServerLoop.log

rem *** Clean-up directories and run TeleFlow Server
rem ***
%1
cd \%2

rem * Create one-time back-up of all crash-log files
rem *
rem mkdir %1\%2\log\backup
rem xcopy %1\%2\log\*. * %1\%2\log\backup\ /Y
```

```
rem * Clear the log directory for the next run of the lines.
rem *
rem del/q %1\%2\log\*. *

rem * Start tfserver...
rem *
cls
echo _____
echo.
echo TeleFlow Server Running - DO NOT INTERRUPT!
echo _____
echo.
TFServer.exe

rem * If TF_Server exits, the following code will run
echo _____
echo.
echo ** TeleFlow Server Error! ***
date/t
time/t
echo _____

rem * Change the title so we know it crashed
rem *
title TF Server Loop - YELLOW FLAG

rem * Take a 30 second rest, then restart
rem *
echo.
echo Re-starting in 30 seconds...
echo.
NtSleep /Q 30

goto LOOP
```

Another batch file, one used to start the above, might look like this:

```
@echo off
rem +-----
rem | TFServerStart.BAT
rem | Manages the re-starting of the TeleFlow Server
rem +-----

cls
echo _____
echo.
echo Starting TeleFlow Server
echo Please wait, or press CTRL+C to abort
echo _____
echo.

C:
cd \TeleFlowApp

echo.
```

```
echo ----- Start! -----  
  
start "TFServer Loop" /min bin\TF_ServerLoop C: TeleFlowApp  
  
echo.  
echo ----- Done. -----
```

## Hang Up Supervision

### Question:

My application seems busy playing the welcome prompt, but there are no calls. When I restart TeleFlow Server each line answers but never hangs up.

### Answer:

This is almost certainly a hang up supervision issue. There are only a couple of possible solutions to this.

They are:

- 1) Have hang up supervision provided on those lines. (Ask whoever provides them about doing this)
- 2) Modify your application to have a limited number of retries on every option/menu or get tt string. (See any of the sample applications packaged with TeleFlow, and how they use MenuSet and MenuMan/InputMan for this) The idea here is that if a caller isn't entering in touch tones and interacting with the system, then they must be gone (or possibly having some difficulties with the system). Either way, the best option is to hang up your self (in the application) to free up the line for the next caller.

## Language Support

### Question:

Our IVR application will need to support English, Spanish, and very soon, French. I understand the '[Set Language](#)' step, but this only changes the Prompts, i.e., how 12 or 'twelve' gets created/spoken.

How do I point to a different 'scripts' directory where the equivalent voice script files are?

### Answer:

When you determine which language to play, simply set a global variable, such as @LANG, to "ENG" (for English), "SPN" (Spanish), or "FRN" (French). Then, in the voice script manager, make sure that each telephony script entry is set up appropriately. (So, it would likely be Vox\@LANG\Scripts) TeleFlow will parse the variable at runtime. If you also set it up this way in the Simulator directory, it will affect where scripts are played from when using the TeleFlow Simulator, which may or may not be desirable.

For a quick change of the telephony field for all scripts, in the Voice Script Manager, go to Search >> Replace >> Type "\Eng\" in the Search For box >> Type "\@LANG\" in the Replace With box >> Check the "In Telephony recording fields" check box >> uncheck "In scripts and notes" >> click "Replace".

## Mixed Boards

### Question:



Can TeleFlow use a mixed board system, i.e. a system that has a NMS AG4000 and a Dialogic D41 boards installed.

**Answer:**

No, it isn't possible to configure TeleFlow to use a mixed board system.

### **Nortel DMS100 pbx**

**Question:**

What card is needed to work with the DMS100 pbx?

**Answer:**

You need an NMS Communications AG 4040 card in order to support this.

### **Outbound Calls**

**Question:**

I have an application that requires that I originate calls is there a way to do this?

**Answer:**

Check out the "Place Call" step. Place Call allows you to initiate outbound calls.

Typically, an outbound application would start, perform whatever setup operations are required (e.g.: connect to an ODBC data source), lookup a phone number to call, use the Place Call step to dial the number, handle the call, then hang-up and loop back to find another number.

There is example code shipped with TeleFlow. Select Alt+File -> New Application. Go to the "Example" tab and select "Outbound".

### **Print Step**

**Question:**

Is there a step to send a string to the printer?

**Answer:**

Yes and no.

You can use the "Run Program" step to run a command line "Print" command. It will print a file you specify. So, from within TeleFlow, you could use the "Write Text" step to write the file out to a text file, and then use the run program step to print it. Use the Delete File step to remove the file from disk after that.

### **Registration**

**Question:**

We've upgraded our system and did a fresh install of TeleFlow 2005. Can I use the old registration code on the new installation? I'm using the same NMS card and number of ports.

**Answer:**

Yes, you can (and should) use the same registration code.

Note that one of the elements of the registration code is the MAC address of the network card in the machine. So, you should use the same network card, as well.

## Simulator: Recording Error

### Question:

When running the application in simulation modes here is the error message we receive:

```
Jun 03 08:36:13.28: [74] Record
Jun 03 08:36:13.28: Recording "\\acclleo\le$\IVRRecruitRecordings\5284.VOX'
Jun 03 08:36:13.62: mciSendCommand MCI_RECORD failed: 322
Jun 03 08:36:13.62: All wave devices that can record files in the current format are in use.
Wait until a wave device is free, and then try again.
Jun 03 08:36:13.62: FAILURE
```

### Answer:

There must be a sound card in the pc to record in simulator.

## SQL: Fetch Rows

### Question:

If more then one record is fetched in a SQL statement how can we handle these records one by one..?

### Answer:

The SQL Statement step runs your query, and the Fetch step grabs a single row. Use the SQL Statement step to run your SQL statement, then, use a Fetch step to gather a row/record of results. Do whatever processing you want to do with those results, and then run the Fetch step again. When the Fetch step follows the bomb/fail path, you have fetched all of the rows.

You must make sure that your SQL Statement and the Fetch step used to retrieve rows from it both have the same handle number.

## SQL: Stored Procedures

### Question:

Can stored procedures be called in Sql Server through the TeleFlow? If yes, how..?

### Answer:

You can. Use the SQL Statement step to run stored procedures in TeleFlow with or without input parameters.

## String Replacement

### Question:

I need to replace '/' with '\'. Is there an easy way of doing this in TeleFlow?

### Answer:

There is also a template flowchart available within TeleFlow Designer that has the specific functionality you are looking for (I believe).

In the TAP screen, in the Steps Toolbox, go to the "Templates" tab. Select the "FindAndReplaceString" flowchart, and add it to your TeleFlow application.

The 3 parameters for this template flowchart are:

- 1) A TeleFlow variable to modify (containing the string you want to replace a portion of)
- 2) The string to replace
- 3) The new value to replace the existing string, if found.

The flowchart "returns" your variable with the replacement, if any.

## Text to Speech

### Question:

Is text to speech an additional purchase?

### Answer:

TTS is an additional purchase, yes. Please contact sales@engenic.com for assistance.

## Voice Detection

### Question:

Can TeleFlow detect if an outgoing phone call was answered by human, a voice machine, fax machine and/or detect a busy tone, non existent tone, etc?

### Answer:

TeleFlow can differentiate on fax, busy signal, SIT, and wait X rings before giving up.

TeleFlow (automated systems in general) cannot differentiate between an answering machine and a human. However, it can provide you with a measurement of the initial voice length of the answering party. A typical person answers with "Hello", while a machine is more likely to say, "Hello, you've reached etc". If you determine anything over 2 (you may need to tweak this number) seconds to be a machine, you can "simulate" answering machine detection (which is what you get with any answering machine detection). This feature is currently only available on the NMS hardware platform.

## Other FAQs

[TFDesigner](#)

[Dialogic](#)

[TFSimulator](#)

## 11.1 TeleFlow Designer

### Called ID (DNIS)

#### Question:

What do I need to do in order to enable the Called ID with the Wait for Call step?

#### Answer:

Called ID is only available when you have DNIS. DNIS (dialed number information service) information is available only on digital circuits (ISDN or Wink Start (robbed bit signaling) protocols).

In an analog environment, each physical phone line has a particular phone number (which is equivalent to its Called ID). If you know what each port's phone number is, you could use @P1 to provide that number to your application, if it requires it for something.

Primarily, the Called ID is only required if you vary the service provided to the caller based on the number they called. In an analog environment, service is varied by port number/line, as opposed to Called ID.

### Email Address With (or variables containing) the "at" @ symbol

**Question:**

I need to define a global variable such that @email=test@domain.com. I cannot do this because the program doesn't recognize @domain.com. Is there anyway around this? Eventually I want to write @email to a text file.

**Answer:**

Since the @ symbol is used to denote a variable, TeleFlow attempts to set @email equal to 'test' and the contents of the string @DOMAIN, and the string '.com'.

You can override this behavior by using two @ symbols when setting the variable.

So:

@EMAIL = test@@domain.com will work.

### File Copy Function

**Question:**

Our application requires Teleflow to copy file from one directory, rename it and save it in another directory. We can do this if everything is on a local machine but can Teleflow do this for remote drives, i.e. copy file to and from remote drives?

**Answer:**

Yes. TeleFlow's Copy and Delete steps call O/S functions to perform file manipulation. Provided the TFServer.exe program has read/write privileges on the remote drive(s), these functions will work just fine. If they don't, it is a Windows networking issue. Note that you can use UNC paths "\\silo\vmmail\myfile.wav", or flat paths "s:\vmmail\myfile.wav".

### Finding a File

**Question:**

Is there any way to find out if a file exists in Teleflow?

**Answer:**

You have 2 options to handle this:

1) Locate a file: Use the "Scan Folder" step. Scan the directory (eg. 'C:\Vmail') you expect to locate the file in. Set the 'file mask' to the actual filename you are going to attempt to play (eg. 'MyFile.wav'), and set a variable to receive directory count. (eg. '@DIR\_COUNT'). The step should succeed, unless the directory doesn't exist, for example, and return a count to @DIR\_COUNT of 1 if the file exists, or 0 otherwise. Use a Compare step to determine the count.

2) The Play step has a hidden action step on it that you can use set the path (referred to as the 'bomb' or FAIL path) to follow if a problem arises when TeleFlow tries to play the file specified. Right-click the step, and select the "unhide action steps" option. (This is not visible by default, because it is only seldom used) Send the flow from the fail/bomb step to whatever handling you

wish to have in the case of a failed play.

## Get TT String

### Question:

When there is more than one allowable terminator in the Get TT String step, (For example \* and #) how can you determine which of the several terminators was used?

### Answer:

In your Get TT String step, include the termination digits \* and # in BOTH the "Tone tone(s) to terminate entry" and the "Allowable touch tone(s)" property. The termination key used will show up in the resulting string ("Variable to receive touch tone(s)") as the last character.

## Get TT String Playback

### Question:

Can I playback a TT String that was entered, playback, and then give the opportunity to accept or re-enter?

### Answer:

Yes. The Get TT String step requires that you supply a variable to capture the Touch Tones. After the Get TT String step, you can use the Say step to playback the digits entered. On the properties for the Say step, set the "Format to speak" field to "Digits", and the "Value to speak" field to the variable used to capture the TT String.

## Initiating Application By Caller ID (or ANI, where available)

### Question:

I have numerous applications on numerous 1900 numbers. Can Teleflow run specific application/script based on the incoming Caller ID/ANI meaning that each application initiated should be specific to the incoming Caller ID/ANI. How do I do this?

### Answer:

We use this often in T1 applications. The simplest form of this would include a wait for call step followed by compare steps with the specific @CALLER\_ID that you are looking for. The true path for each of these compares could go to separate tams of the same application. More effective yet, you could use the Run Application step, and run separately written applications, each of which is run from one TeleFlow Application that waits for calls, determines the application to run, and runs it.

## Interruptible Play File Skips Next Play on Touch Tone

### Question:

I have my initial greeting set as NOT INTERRUPTIBLE! But when I test it in a live environment I am finding that if I press a TT during that message it skips the next PLAY step before the MENU step. It finishes playing the initial greeting but not the next.

### Answer:

The interruptible setting on a Play step simply indicates whether that play should be terminated by a Touch-Tone. It does not control the Touch-Tone input buffer, which will still gather the TT input and save it for later processing. In order to achieve the functionality you want, place a Clear step immediately after your initial greeting.

## Port to Port Communication

### Question:

We're developing an application that requires the following.

Caller on Port 10 records a message for caller on port 15. How can we specify TF to play a "incoming message" prompt to caller on port 15 and playback the recorded message if he accepts.  
Note: Only caller on port 15 will hear the incoming message prompt and get the message.

Specifically, how to playback at a specific port?

Can we initiate a call switch on any 2 ports so that callers can chat live?

### Answer:

My recommendation would be to use a database table. Have port 10 write an entry into the table, and port 15 would query the table looking for any entries "addressed" to it. The same functionality could be implemented by writing and reading temporary files on disk.

## Switch Ports Together

### Question:

Can I switch 2 callers who are already in the system together so that they can have a live conversation? Do I just do switch port X with port Y?

### Answer:

You do just switch port X with port Y. Use the Switch Call step. The first port ("port X") is the current port that your application is running on. The step properties has a field, "Connect current port to port number..." to specify the other port.

Only one of the ports needs to execute this step. The other port will be switched in automatically. The ports will stay connected until the port that initiated the Switch Call step, executes a Switch Disconnect step.

After the ports are connected together, it is best to put each TeleFlow application into a loop, waiting for a hangup event. A Play step to a Connect Point step, back to the Play step will do the trick. Be sure that the hangup handler for the application that made the connection includes a Switch Disconnect step.

Examples of a this functionality are included with TeleFlow. By default they are installed at:  
C:\Program Files\TeleFlow\Templates\Example\TransferDemo

## Variable case sensitivity

### Question;

Are variables case sensitive or not (e.x DBLOOP same as dploop)

### Answer:

TFServer reads all variables as UPPERCASE. @Variable is the same as @VARIABLE.

## 11.2 Dialogic

### Vox Files

In TeleFlow, VOX files are (currently) the only supported audio file format for Dialogic. You can use the "Run Program" step in TeleFlow and use TFSoundTransformer.exe to convert files from VOX to WAV and back.

To do this, get the latest TFSoundTransformer...

Follow the link below to ensure you have the most current version.

<http://www.engenic.com/TFSoundTransform.zip>

In the zip, you will see some examples of how to run TFSoundTransformer. It would be best to test the transform in a CMD.exe window to make certain it does what you need it to. TFVoicePlayer (also included in TeleFlow can be used to listen to VOX files). You should be able to use some trial and error to create what you need. Once done, take the command and enter it into the Run Program step. Make sure that TFSoundTransformer is in the path or in the same directory as TFServer.exe

## 11.3 TFSimulator

### Automatically Loading Application in Simulator

#### Question:

Can I automatically load an application into the Teleflow Simulator.

#### Answer:

There are actually a number of parameters required.

#### You will need to structure the call like this:

```
TFSimulator.exe /DX:\ 1000 "Standalone" "C:\TELEFLOW\TFDemo.TAP" 1
```

Where:

/DX:\ = Something very important

1000 = Also something important

"Standalone" = Title on the program

"C:\...TFDemo.Tap" = App to run

1 = Passed parameters (generally port number)

#### The order of the parameters is:

##### 1. Application Path

A path prefixed by -D (must be enclosed by quotes if there are spaces)

e.g. "-D:C:\Program Files\TeleFlow\Applications\AnswerTest"

##### 2.Application ID

A numeric value that is ignored but still necessary

e.g. 1000

##### 3.Application Name

A text name for the application (must be enclosed by quotes if there are spaces)

"IVR Line 1"

#### 4. Application File Name

The pathed file name for the application (must be enclosed by quotes if there are spaces)  
e.g. "C:\Program Files\TeleFlow\Applications\AnswerTest\AnswerTest.tap"

#### 5 to 13. Application Parameters @P1 to @P9 (optional)

The values that @P1 to @P2 should take on

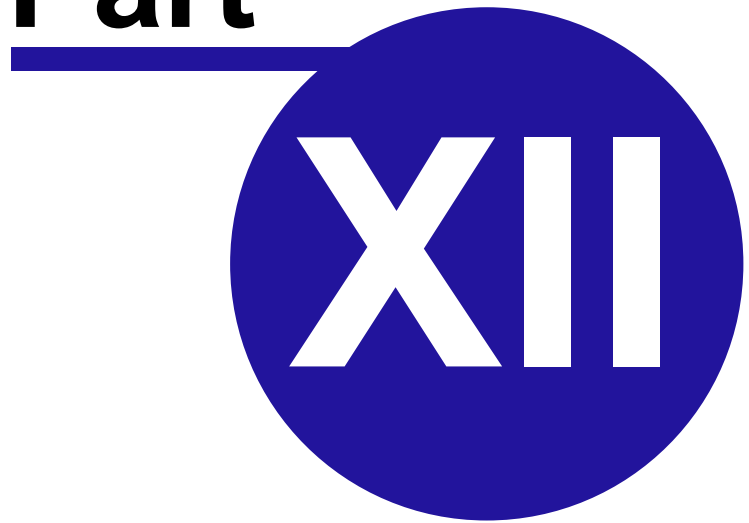
Here is an example of a how to start TFSimulator with parameters:

```
TFSimulator "-D:C:\Program Files\TeleFlow\Applications\AnswerTest" 1000 "IVR Line 1"  
"C:\ProgramFiles\TeleFlow\Applications\AnswerTest\AnswerTest.tap" 1
```



**TeleFlow**

**Part**



## 12 Glossary

TERM	DESCRIPTION
<b>Events</b>	See the 'Data Collected' subsection of the 'Statistics' section of this document for information on 'Events'.
<b>Fatal Error</b>	In TeleFlow, a fatal error occurs when a key piece of data required to successfully run the step is missing (Attempting to compare a variable that doesn't exist, for example), when something fails and the failure is not or cannot be handled, or when a path from one step to another is broken. An TeleFlow Application programmer can also specifically generate a fatal error using the <a href="#">Fatal Stop</a> step.
<b>Hunt Group</b>	<p>This is a telephone company term. A hunt group is any number of Telephone Lines grouped together under a Pilot Number. When a call comes in, if the pilot number is busy, the next number in the group is tried. If that line is busy, the next number is tried, and so on. Hence, the group "hunts" for an available line to accept the call.</p> <p><i>See also "Telephone Line Group" and "Pilot Line"</i></p>
<b>IVR</b>	Interactive Voice Response; a generic term for the technology that enables computers to be used to handle telephone calls and interact with callers.
<b>IVR Port</b>	Term used to describe a single phone line connection to an IVR system. In this case, a "single phone line" is capable of supporting a single "conversation" between two parties (such as a caller and the automated system/IVR)
<b>Line</b>	This is an ambiguous term, and should be avoided. A "Line" could refer to an IVR Port, a physical telephone line, or a single instance of a TeleFlow application. One instance of a TeleFlow application monitors a single IVR Port, which is connected to a single physical telephone line. Hence, the application itself is often referred to as a "Line".
<b>Pilot Line</b>	<p>A single telephone line designated as the main phone number for a "Hunt Group".</p> <p><i>See also "Hunt Group"</i></p>

---

<b>Speech Recognition sever software (SR)</b>	This software provides the ability “comprehend” or “understand” human speech. Grammars are used to compare voice recordings (called Utterances) from callers or users of “Voice Applications” including those created in TeleFlow. Utterances when matched to a grammar entry create codes or tests, which can then be interpreted and acted on by the Voice Applications Software. SR servers generally communicate to Voice Applications Software (Such as TeleFlow Server) using IP packets to accept vocalized waveforms, and then return words and/or text.
<b>Supporting Software</b>	The software that provides additional functionality for the Voice Applications Server. Software in the class may include web server, data management and loading applications, and reporting solutions.
<b>Text to Speech server software (TTS)</b>	This software provides the ability to vocalize text or words. This software will create human sounding speech best on dictionaries for common words, names and places. For other words, the best guess is taken based on sounds within the human language. TTS servers generally communicate to Voice Applications Software (Such as TeleFlow Server) using IP packets to accept words and text, and then to return the vocalized waveform.
<b>Telephone Line Group</b>	A logical grouping of Telephone Lines to serve a common purpose. A Telephone Line Group will usually have one main telephone number, or “Pilot Line”.  <i>See also “Hunt Group” and “Pilot Line”</i>
<b>TeleFlow Applications</b>	Voice applications created using TeleFlow’s graphical interface known as Designer, are saved to a tagged file format. These tagged files (including “Tap” and Tam”) are provided to the TeleFlow Server Software, which then processes them according to rules and logical flow. With TeleFlow, these Voice Applications are expressed by flowcharts.  Functional icons, known as “Steps”, are used to not only build logical flow, but also interact with other environments such as databases, Text To Speech (TTS) servers, and process results from Speech Recognition (SR) servers.

**TeleFlow Server Software**

The TeleFlow Server Software accepts tagged files (including “Tap” and Tam”) created with TeleFlow Designer. TeleFlow Server executes the instructions in the tagged files. TeleFlow Server communicates with voice hardware, databases, and other services such as Text To Speech (TTS) servers, and Speech Recognition (SR) servers. This activity specifically comprises an operating Voice Application, or “TeleFlow Application”

**Touch-tone Buffer**

Whenever a caller pushes a key/button on the phone, this generates a touch-tone. In TeleFlow, these touch-tones are accumulated in a buffer until a step that processes touch-tones is encountered. A given step may remove one, several, or all tones from the buffer. While the buffer has one or more touch-tone in it, TeleFlow will skip any Play or similar step unless the step is specifically set to be uninterruptible.

**Utterance**

An Utterance is a vocalization from a caller. It could be anything including a simple word, a short sentence, or even a cough. TeleFlow Server, when configured as a Speech Recognition environment will sense sound, and record it to memory. It is then passed to the Speech Recognition (SR) Server environment and tested for matches in the grammar currently in use.

**Voice Applications Server**

A Voice Applications Server, in the computer that houses the voice hardware and runs all the software required to provide Voice Services. This documentation may sometimes also refer to this as the “TeleFlow Server Computer”.

**Voice Applications Software**

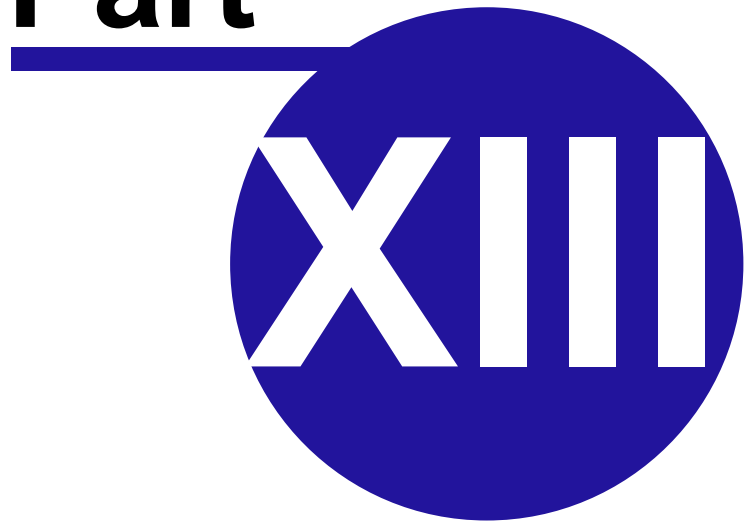
This is the software and/or applications that provide voice services. Voice Applications Software includes TeleFlow Server Software, and the applications (created using TeleFlow Designer) on which it runs.

**Voice Hardware**

Hardware that provides the connection to Public Switch Telephone Network (PSTN) and Voice over Internet Protocol (VoIP) networks. This hardware enables TeleFlow Applications to provide the same or similar service for all communications protocols. Separate hardware is required from a PC computer because it is specifically built for the task of processing voice calls.

**TeleFlow**

**Part**



## 13 Appendix 1 - Speech Recognition

This document provides points to consider and a step by step flow through for developing Speech Recognition (SR) environments in TeleFlow. The following resources and materials will be required for development of a speech system

- [TeleFlow Designer](#) – Development Environment
- [TeleFlow Server](#) – Production Environment
- [Nuance / Lumenvox](#) - SR processing environment on both Development & Production
- [TeleFlow SpeechTrainer](#)

Brand new SR system may require significant adjustment to the grammars and voice work in order to function effectively. Training to account for the different ways in which people speak and communicate will need to be taking into consideration. No matter how carefully a system is planned, people will always interface with it in different ways than anticipated. The user experience can then be gauged using TeleFlow SpeechTrainer. Feedback viewed and listened to through SpeechTrainer can help finalize the grammars and provide insight into how the application be modified to help ensure a good experience by most callers.

Speech Recognition systems are very difficult to create and require careful planning. The following points will help you to plan development and train the application to function correctly over time

### 13.1 Planning and Development

The following notes provide some thoughts that can be used to determine how the planning and development process should take place.

Please note that SR development is no trivial matter, and that the process will require input from many people over several weeks or months.

- Make sure the right people are there... the people who
  - Listen to what callers say (call center operators)
  - Marketing people (what the jargon should be etc)
  - Managers
  - Technical people
- Ensure to educating the speech team on the methods that will be employed to build the system. Determine the plan ahead of time, and share with all involved.
  - Provide examples of what is reasonable for use.
  - Explain that SR is not “StarTrek”.
  - Intelligence is not involved. Anticipation and caller guidance is the key.
  - SR It is not Artificial Intelligence. This is a common misconception.
- Get all user requirements clearly stated before creating design]
- Be careful with prompts using the word "or"
  - “or” tends to be answered with a Yes or No, and can be confusing.
- Prompts and Grammars must be aligned
  - Do not change the prompt half way in, or the grammars will need to be updated
  - Make certain all prompts have been defined early on
- Don't skip iterative usability testing
  - Make certain to test various elements of the design independently
  - Menus
  - Error handling

- Processes  
Used to ensure the speech system works with the callers mental process
- Don't shorten testing tasks
  - Make sure testing tasks are not done in parallel.
- Test on Production and not Development
  - A system should be in production for users to test against
  - Development should move save points to Production
  - Testing on Development means testing against a moving target
- Provide proper instructions for testers to call with
  - A specific document with numbers to call, and test accounts etc
- Perform evaluative usability before the system is ready
  - Evaluate as the system is being developed

### **Production Walk Through**

The following meetings and steps need to be taken from start to completion.

- Requirements Analysis Meeting
- Gather user requirements
  - Ensure correct people in room or in conference
  - Use sticky notes on white board (or net equivalent)
- Define business requirements
  - What needs to be achieved
  - What data elements need to be gathered
- Identify application requirements
  - A functions list should be created by the end of meeting
  - Ensure front line staff have equal input, or important information will be lost
- Design Documentation
  - Persona and audio design
    - How do you want the system to sound
    - Consider tone and use of language
    - Decide on voice talent
- Dialog design
  - Create scripts around functions list
  - Create questions list
  - Gather list of Grammars that will be required.
  - Run questions by a test group, and gather addition words (answers)
- Application design
  - Database Dictionary describes Data Elements
  - Provide Final Design Document
    - Scripts
    - Grammars
    - Functions
    - Interface to database

- Other programs (web / configuration / reports etc)
- Client Sign off
- Development and Implementation
  - Persona and audio production
  - Grammar development
  - Application development
- Testing
  - Application tests
  - Recognition and traversal tests
  - Evaluative and usability tests
- Tuning and Monitoring
  - Pilot tuning
    - Post deployment tuning
    - Post roll-out monitoring

Once the last step is complete, the system will be ready for general use.

### **Avoid Common Errors**

The following list provides an indication of common errors.

- Make sure you have people who know how to write speech systems.
  - This can save time over trial and error development.
- Would you like “Fries or Salad”? // User will say “Yes”
  - s/b Which would you like... Fries (pause for answer), or Salad?
  - Better yet, stay away from “or”
- Avoid “Please say your phone number”
  - s/b What's your phone number?
  - Consider allowing touch tone entry for prompts like this
- Choose very different words to ensure best results from the recognizer
- Keep grammars short, and then only expand on pilot testing when users speak words
  - Use pilot testing to gather words that users speak
  - Users will often speak a different set of vocabulary than expected
  - Use the SpeechTrainer to review the words callers speak.
- Make sure you design for the callers, and not for the sake of speech
  - Touch tones may be more accepted than speech
- Do testing and development on entirely separately systems
  - Be sure to test on the production system, and not the demo system
- Develop a test matrix to cover all aspects of testing
  - Be sure to balance testing across entire system
  - A common error is to test certain sections, and assume the rest will work
  - For this reason, it is a good idea to test each functional section separately
  - Create a document and check off when tested



- Make sure you have detailed instructions on how callers (participants) should test
  - Test information such as account numbers should be apparent
  - Consider that callers testing have different intentions than your client callers
- Check system to make sure it covers all original user requirements
- Loud locations will sometimes trip SR. Test the system from a variety of locations
- Cell phones (and some VoIP services) are not as clear as land lines.
  - Test with a variety of telephone handsets
  - Provide routes to touch tone enter certain information where readily possible
- Avoid pretending to be human
  - The caller should never hear the words “I’m sorry...”
  - The caller should not hear the same “We did not receive a valid response” error
  - Try cycling through five or six messages to keep the annoyance factor low
- Make sure the callers are interviewed at the end to ensure they like it
- Make sure a reasonable amount of time is set apart for the testing
  - Be certain to review each day’s results and tune (SpeechTrainer) each day
- As your callers train the system, the system will train them overtime.
  - Seasoned users learn to speech in a way the system understands
  - Before roll out, ensure a new set of users are able to adopt the system
  - Provide any user documentation in final form, if applicable.

### **Problems and Inconsistencies**

Numbers with a specific number of digits are much easier to recognize over variable length ones. When we know we are waiting for seven digits (for local telephone numbers example), the caller can speak the numbers at any pace they want to, and the system will simple wait until it hears all seven... within reason.

Variable length numbers are exceptionally difficult to account for, because timing is everything. If waits between numbers are too extended, then numbers can get cut off when the system thinks the caller may be finished speaking.

Both variable and fixed numbers should be “chunked” where possible to guide the caller on how to speak. North American telephone numbers are generally expressed as 604-555-1234, and callers will generally speak them as “six”, “oh”, “four”, pause, “five”, “five”, “five”, pause “one”, “two”, “three”, “four”. This chunking can be useful to program the system for number utterances when callers leave very long spaces between numbers.

In cases where numbers are variable length, see if it is possible to manufacture a fixed length number, or account for numbers by chunking them. For example, “1” through “99999” could become, “100-001” through “199-999”. If at all possible, try to create numbers that chunked in a familiar manner. Seven and ten digit numbers in North America are commonly spoken, and accordingly other numbers of the same length are commonly spoken in the same chunks.

Cell phone usage is prevalent today, and requires some additional handling in loud areas. Where possible, provide an opt-out when entering numbers so that Touch Tones can be used. If a grammar is worded correctly

Caller frustration (expletives) is something that you may want to account for in grammars. However, as where some callers realized that pressing “zero, zero, zero” repeatedly in touch tones systems brought them to an operator, the same applies to spoken voice. Expletives spoken repeatedly may also be used by some callers to by pass automation.

### **Challenges Speech Recognizers Contend With**

#### **How People Speak:**

Accents, gender, age, mannerisms, speed, inflection, volume and energy all contribute to the variety of ways a word or phrase can be spoken. Developing a Speech Recognizer to understand most people, most of the time is complex and involved task.

#### **Inferring Meaning:**

Speech Recognition applications must be developed to listen for words they are expected to hear at specific times in the conversation. Because computer systems don't yet have the ability to comprehend human language, they cannot infer what was probably being requested or stated. Specific word sets need to be defined in grammars that can then be interpreted against. For these reasons, the law of declining returns very quickly applies itself to complex speech applications. Accounting for the many spoken possibilities at any time also means that the costs of training the system may grow incrementally.

#### **Background Noise:**

Because of the nature of cell phones, voice quality is not as good as land lines. In addition, cell phones are often used in traffic or out in public where there is additional background noise which can drown out the voice of the caller. For this reason, Speech Environments are unable to recognize utterances from cell phone users some or much of the time.

#### **Continuous Speech Recognition:**

Because computer systems are unable to understand human language, they are unable to infer true meaning. For this reason, continuous speech recognition is exceptionally difficult to do, except in the context of a very specific application. The classic example that is often provided is the phrase "to recognize speech" which sounds similar to the phrase "to wreck a nice beach." Because people don't speak words independently from one another, computer systems are unable to understand separate words in a sentence.

### **Expectations of a Speech Enabled System**

Reasonable expectations from an SR system include:

- Low cognitive load for the caller (Intuitive design helps adoption)
- Efficiency (No long menus)
- Graceful error recovery (Sounds humanistic, without pretending to be human)
- Accuracy
- Clarity

Unreasonable expectations include:

- Perfect recognition of spoken numbers and commands

□ – Provide an ideal response for every caller's voice requests

**IMPORTANT:** A common misconception about SR is that it will understand what is being spoken. Sounds are identified against a dictionary, which then creates a word. All functionality must be specifically developed by programmers to account for expect situations. A friendly voice can help to guide the caller through.

## **Delivery of a Speech Enabled System**

### **Recordings**

The voice actor or actress you choose must be well thought out, and match the general concept of the voice system. You should also expect to have several recording sessions as the voice can have a strong impact on acceptance of the system by testers and ultimately, the client.

When the voice actor or actress records, ensure the tempo is at a reasonable pace. This is important as slow recordings may prompt the caller to speak slower for SR. The pace of the recordings can act as a guide for the caller.

### **Client**

From the start, it should be explained that SR is not "Star Trek", and that the expectations should be in line with the investment of development.

### **Testers**

Try not to education testers too much ahead of time, and only provide the basic information that client callers should have (or may not have) as the case may be.

As your testers use the system, they will learn to use it more effectively as time goes by. Effectively, they are trained by the system over time. Unless this is the purpose of the system, a separate set of test users should be kept aside for late testing, and for interview on the voice work etc. Systems that are intended to be usable by first time callers will need testers who are entirely unaware of the capabilities of the system.

### **Feedback**

Very soon after the system is tested, a conference call and/or questionnaire should be arranged and/or filled out. Be careful to keep the conversation to questions only, and listen to feedback. A conference may be ideal, as testers may help each other remember aspects of the systems they had difficulty with. If the testers are employees of the company accepting the system, consider keeping the conference call to just the testers and not their managers.

### **Other Means for Review**

Call recording (inquire about TeleFlow CallCapture), can be a great way to gauge the user experience. Call recording systems record the entire call so that you can hear exactly what the caller heard and said. Leaving the system in place can be a very good way to spot check over time. Please note that if you employ a call recording system to review calls, that a message stating "This call may be monitored for quality assurance" may be necessary according to law.



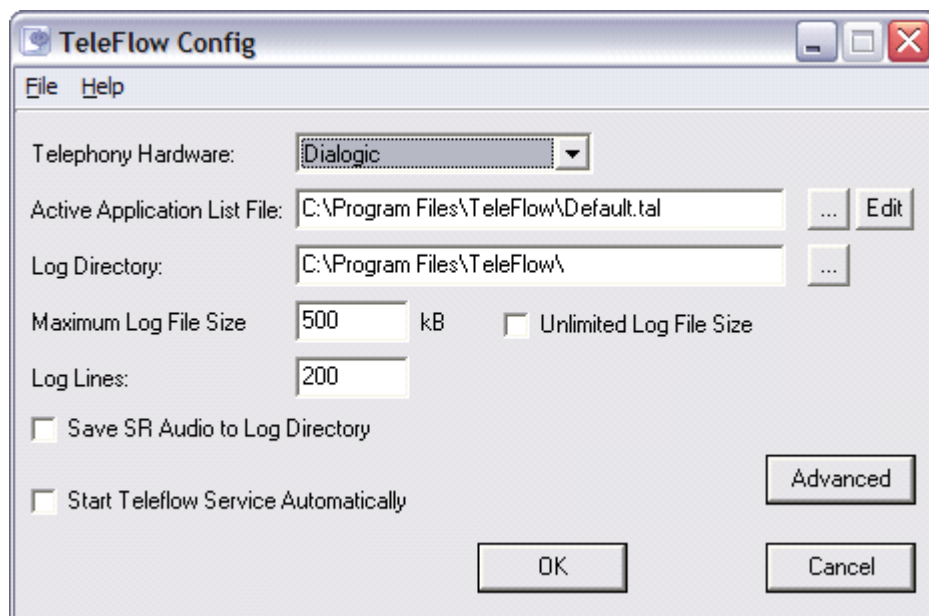
### **TeleFlow SpeechTrainer**

TeleFlow SpeechTrainer allows you to review recognitions, and determine what happened with all of the recognitions on a call by call basis. Then you can make adjustments as needed to how you prompt your caller, and how information is requested from the caller.

Speech training is a process comprising many things, such as modifying grammars, the way in which the caller is prompted, adding confirmations where callers are having trouble, etc.

### **Setting Up the TeleFlow to Log Speech Recognition Events**

Run TFConfig and check on "Record Speech Recs" click "Save SR Audio to Log Directory"



Restart TeleFlow Server to accept the new setting. Each day a new folder is created of Speech Recognitions and related files. These are files that are created when callers are speaking to provide voice commands to the recognizer (ie: Nuance SR / LumenVox SR)

### **Transferring Data to the SpeechTrainer Computer**

SR files will create on a daily basis. At the start of the day, the previous day's SR should be reviewed. You can review the files on the TeleFlow Server computer if it has a sound card. Generally however, SR files will be reviewed on another desktop computer.

Log files are:

- Generated once per day per port
- Contains information about only those recognition results flagged by the application via the SR Flag Result step (refer to Designer for this step).
- Contains comma separated data, just like a .csv file.

**Location:** {log}\ ST{date}\  
where: {log} = TeleFlow log directory

{date} = The current date yyyyymmdd

**File name:** ST{date}-{port}.FLG  
where: {date} = The current date yyyyymmdd  
{port} = The port number

To bring back those files, copy the needed log file (.Log) into the that days folder as described above, and copy the entire directory (folder) to the client computer.

### Using SpeechTrainer Interface

Install the SpeechTrainer software on the computer you will be using to review the SR output by. This program is a separate install, and must be installed on a computer that is running Windows XP, 2000 or 2003 with a sound card.

- Start SpeechTrainer
- Click the "Open CSV" button, and choose the SR.csv from the directory
- The left hand pane is each complete call that was made. The right hand pane shows the speech recognitions that were recorded.

When you click on a line in the SR pane the details for that SR are detailed below it.

If the previous play to the SR is the same as the last heard play then only the previous play details are populated.

When you click on the flag button, a copy of that SR is placed in the SR dir\Flog for later use. Also the number row of the SR is marked yellow.

If you press play for the "Previous Play" or "Last Heard Play" and the file does not exist to the absolute path, then a dialog will open. Once you choose the location of the file, every other play will attempt to look in that directory as well.

Show log will open the log file, as long as it has been moved into the SR directory. Larger files can take a few seconds to open. Once the log file is up, you can press the find button. This find dialog will be populated with the name of the SR.wav.

Display failed Speech Recs Only: will only display recognitions where the SR Result Confidence is less than Threshold Confidence.

### Improving SR Applications with SpeechTrainer

The SpeechTrainer will help with several areas. Determining what the caller is asking for at which point in the system, and pronunciation are the primary benefits of using SpeechTrainer.

SpeechTrainer will allow you to listen to the responses (or utterances) from the callers. By reviewing the utterances, you can determine what the expectations of the caller were, or if they were speaking commands that were not accounted for.

Grammars can then be adjusted and provided back to TeleFlow Server for the next days testing or usage.

SpeechTrainer will also help determine the style of the application. For example, occasionally a critical piece of information is requested by the system. During these points you may need to restate what was heard so that you can be certain. You don't need confirmations everywhere, but you can use them for specific areas you are concerned with, or where difficulties arise. So, if in

speech training, you do find that "Option A" and "Option B" don't work as well as you need them to, confirm them with the caller, like this: "Did you say Option A?" and retrieve the "Yes" or "No" response, which has very distinct and dependable results.

SpeechTrainer will not specifically indicate how to solve a problem, but will provide insight into where callers are having trouble.

New systems should be reviewed on a daily basis in order to improve grammars. Overtime, reviews can be reduced to once a week once stabilized.

I:\Instructions\Speech Recognition Development\SR Development with TeleFlow SpeechTrainer.doc

## 13.2 Tuning Speech Recognition with TeleFlow SpeechTrainer

The following topics explain how to use the TeleFlow SpeechTrainer to tune speech recognition applications developed in TeleFlow.

[Starting with SpeechTrainer](#)  
[How to get Speech Recognition information](#)  
[Recognition Flagged Results file](#)  
[Recognition Results file](#)  
[Summary File](#)

### 13.2.1 Starting with SpeechTrainer

#### Starting with SpeechTrainer

- Start by opening a SR directory.
- Click on the Open CSV button.



- Each port will have a **.csv** (comma separated value) file.
  - The file will be named, for example: st20051112-3.csv, which would be port 3 for October 12, 2005.
- The date of the file will be displayed next to the Open CSV button. The pane on the left shows each individual call for that port. The pane on the right shows each SR for that call. The information includes:

<b>Calls</b> Left Panel	
#	Call number
Start Time	The time of the call start in hh:mm:ss.sss
Port	Port of the call
Recs	The number of recognitions for this call
Secs	The seconds since midnight when the call started.

<b>Speech Recognitions</b> Right Panel	
#	Speech Recognition number
Conf%	Confidence percentage that the speech recognition returned
Adjusted	Final result that the speech recognition returned
Start Time	The time that this SR started
End Time	The time that this SR ended
Length	The length in seconds and milliseconds of the SR
SR File Name	The file name of the recorded SR file
Heard	What the speech recognition first heard
ExecResult	Most recently executed Play step result, "COMPLETED" or "INTERRUPTED"
PlayedRes	Most recently played Play step result, "COMPLETED" or "INTERRUPTED"
RowID	The row in the csv where this SR is found.

- Once a Speech Recognition in the left panel is clicked on, which will make the row blue, the lower panel is populated with that row's information.
- The wave form section shows what file is being played. By default the speech recognition wave file is loaded automatically and played.
- You can click on the wave form and play from anywhere in the form.
- The lower pain contains the Last Heard Play, Previous Play and Speech Rec.
  - If the last heard and the previous plays are the same, then the Last Heard Play is grayed out. The previous play is the last play in TeleFlow before the speech recognition. If there were a number of plays before the speech recognition and the caller barged in on any other than the "previous play" the one that was interrupted would show up as the Last Heard Play.

### Speech Recognition highlighting

There are 3 possible colored fields in the Speech Recognitions panel.

- If # is yellow then the user has Flagged the SR.
- If Conf% is red the confidence percentage was below the threshold confidence, which is considered a failure.
- If Adjusted is yellow TeleFlow marked the SR as flagged.

You can choose to show only failed and/or flagged speech recognitions. To the right of the Open CSV button there are check mark boxes for

- Display Failed Speech Recs
- Flags – either user or TeleFlow noted.

### User Flagging Speech Recognition



If the "#" in the Speech Recognitions panel is yellow, and the SR is flagged by the user, a directory is created in the open SR log directory named Flag. A csv is created containing:



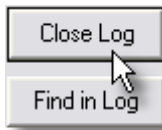
- Name - the name of the SR wav
- Heard- what the speech recognition returned
- Adjusted - what the speech recognition adjusted the returned value to
- Conf - the confidence that the speech recognition returned
- Thresconf - the threshold confidence that was set in the application.

Each of the \*.wav files that are flagged are also stored in the Flag directory.

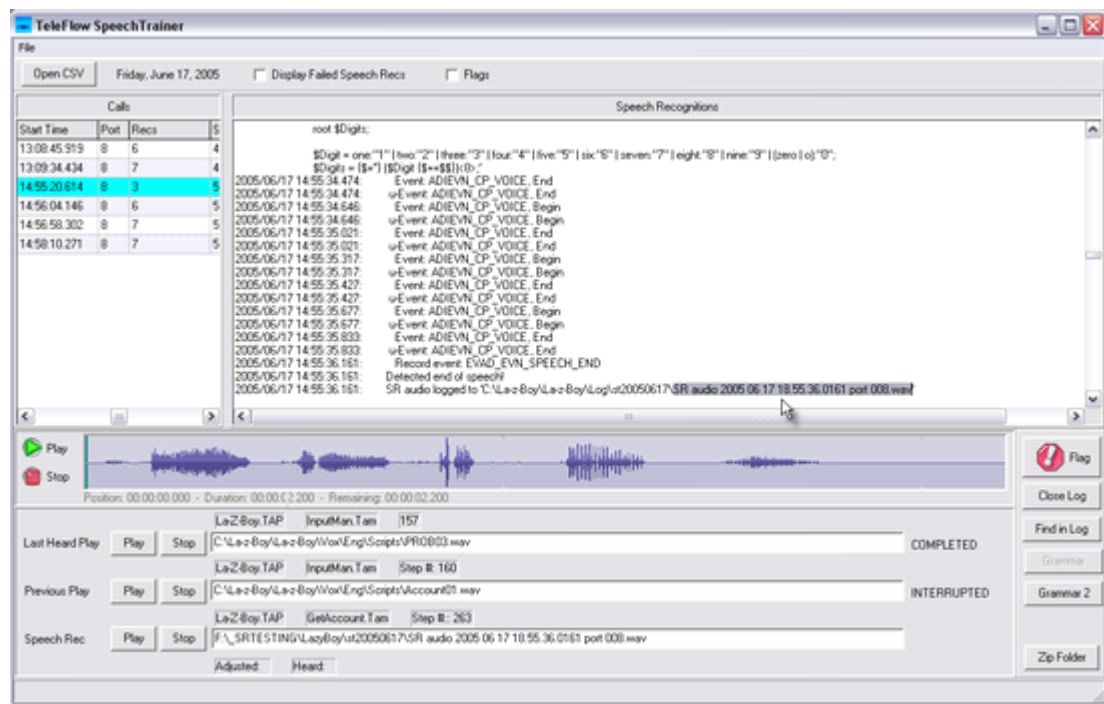
### Showing the Log



When the **Show Log** button is clicked the Speech Recognition panel is replaced with the text of the log. The **Show Log** button is renamed to **Close Log** to return to the Speech Recognition panel.



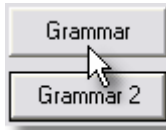
The **Find in Log** button is populated with the name of the speech recognition so that part of the log can be shown.



If the log file is not found, make sure to move the log file into the opened SpeechTrainer (e.g. st20051112) directory.

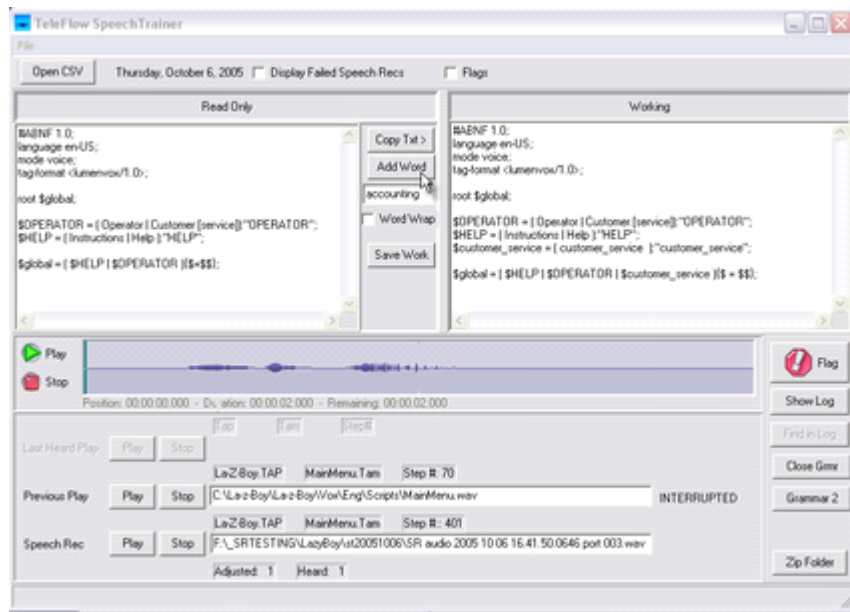
## Showing Grammar used in Speech Recognition

To show the grammar that was sent to the speech recognition click on the **Grammar** button.



The panels Read Only and Working are shown. The Read Only panel shows the grammar that was used.

The example shown was sent to [LumenVox](http://LumenVox) using standard SRGS grammar.



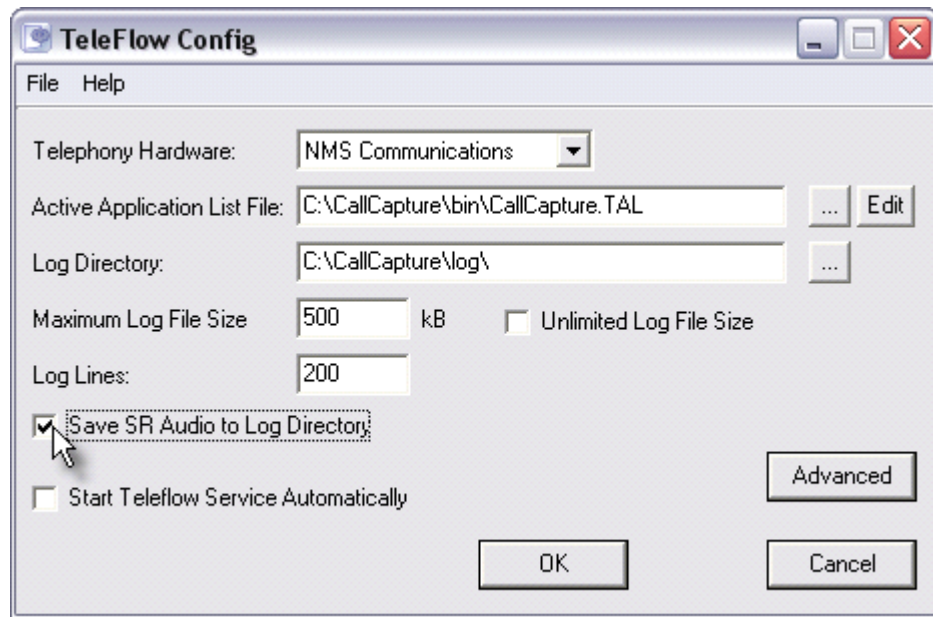
To quickly add a word to the grammar, start by clicking the Copy Txt button. This will populate the Working panel. Next enter the word you wish to add in the edit window below the Add Word button. To finish, click the Add Word button and then Save Work to the location of the grammars the TeleFlow application is running.

Some speech recognition software allows a second grammar to be run on the same file. In this case the Grammar 2 button will be populated.

## 13.2.2 How to get SR information

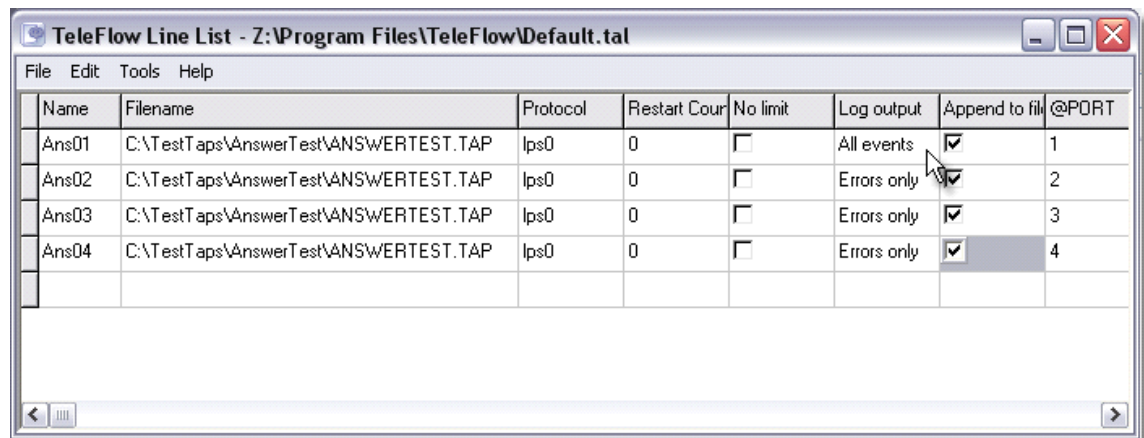
### How to get speech recognition information

[TeleFlow Config](#) (TFConfig.exe) must have the setting “Save SR Audio to Log Directory” set to on. When this is done with a speech recognition system, Speech Recognition (SR) files are saved in the specified log directory.



Each day a new directory of SR files is created. If the date was October 12, 2005 (2005-11-12) the directory name for that day would be st20051112.

To help with logging information all logging should be turned on for the application. To do this change the Your\_Application.TAL file from logging errors only to log all events. Change all the ports that will be producing speech recognitions to log all events.



Once the SR files have been created during testing move the needed \*.log files from the log directory into the SR log directory. To open log files in Speech Trainer the log files need to be placed in the SR directory being read.

### 13.2.3 Recognition Flagged Results file

Generated once per day per port

Contains information about only those recognition results flagged by the application via the SR Flage Result step.

Contains comma separated data, just like a .csv file.

**Location:**

{log}\ ST{date}\

**where:**

{log} = TeleFlow log directory  
 {date} = The current date yyyyymmdd

**File name:**

ST{date}-{port}.FLG

**where:**

{date} = The current date yyyyymmdd  
 {port} = The port number

**Data:**

#	Field	Description	Example
1	CallStarteDate	Date call started, TeleFlow format	2005/03/09
2	CallStartTime	Time call started, seconds since midnight	57882.570
3	Port	Port number	1
4	SRAudioFile	Audio file name where SR input buffer recorded (name only)	SR audio 2005 02 14 19.54.53.0573 port 001.pcm
5	SRHandle	TeleFlow SR result handle value (integer)	18

**Notes:**

- Fields **CallStartDate**, **CallStartTime**, and **Port** (1-3) form a unique call identifier. Data from multiple files (ie: multiple ports, but same day) should be merged and sorted by these unique identifiers, so SR data is reviewed in call order.
- Field **SRAudioFile** (4) corresponds to the **SRAudioFile** field (37) in the Recognitions Results file.
- Field **SRHandle** (5) corresponds to the **SRHandle** field (43) in the Recognitions Results file. The handle is a 32-bit unsigned integer. This number IS NOT a unique identifier, as each port has its own incrementing number. The number starts at 0 when TFServer starts and will continue to increment for every recognition processed by that port.

### 13.2.4 Recognition Results file

Generated once per day per port  
 Contains information about each recognition processed, one row per recognition

**Location:**

{log}\ ST{date}\

**where:**

{log} = TeleFlow log directory  
 {date} = The current date yyyyymmdd

**File name:**

ST{date}-{port}.CSV

**where:**

{date} = The current date yyyyymmdd  
 {port} = The port number

**Data:**

#	Field	Description	Example
1	CallStartDate	Date call started, TeleFlow format	2005/03/09
2	CallStartTime	Time call started, seconds since midnight	57882.570
3	Port	Port number	1
4	LogFileName	Name (only) of the log file for this application	Ans01.log
5	PlayedStepTAP	Most recently played Play step TAP	C:\BOSS\DNISEval.TAP
6	PlayedStepTAM	Most recently played Play step TAM	Welcome
7	PlayedStepNum	Most recently played Play step number	107
8	PlayedStepName	Most recently played Play step name	Play
9	PlayedScript	Most recently played Play step Script code	Welcome01
10	PlayedFile	Most recently played Play step audio file	C:\TestTaps\La-Z-Boy\Vox\Eng\Scripts>Welcome01.Vce
11	PlayedResult	Most recently played Play step result, "COMPLETED" or "INTERRUPTED"	INTERRUPTED
12	PlayedTo	Length of time that audio played for, in seconds	7.552
13	ExecStepTAP	Most recently executed Play step TAP	C:\TestTaps\La-Z-Boy\VoicePal.TAP
14	ExecStepTAM	Most recently executed Play step TAM	GetAccount
15	ExecStepNum	Most recently executed Play step number	107

16	ExecStepName	Most recently executed Play step name	Play
17	ExecScript	Most recently executed Play step Script code	Account
18	ExecFile	Most recently executed Play step audio file	C:\TestTaps\La-Z-Boy\Vox\Eng\Scripts\Account.Vce
19	ExecResult	Most recently executed Play step result, "COMPLETED" or "INTERRUPTED"	INTERRUPTED
20	ExecTo	Length of time that audio played for, in seconds	0
21	SRStepTAP	SR step TAP	C:\TestTaps\La-Z-Boy\VoicePal.TAP
22	SRStepTAM	SR step TAM	GetAccount
23	SRStepNum	SR step number	118
24	SRStepName	SR step name	SR Get Digits
25	NoVoiceTO	No-voice timeout value	3000
26	EndVoiceTO	End-of-speech timeout value	1000
27	ThresholdConf	Threshold confidence	50
28	RetainAudio	Retain audio on threshold failure; boolean 0 or 1.	1
29	RecognizerParm	SR parameter(s) or package identifier: varies from one recognizer to the next. For Nuance it is a folder name where all SR grammar data is found. For LumenVox it is a file/path reference to a parameter file.	C:\Program Files\TeleFlow\Vocrec
30	GrammarBase	Base grammar used; internal package name, or file name (LumenVox only), or grammar package name (Nuance only)	Internal:DIGITS
31	GrammarBaseData	File name containing the grammar text copied from GrammarBase (Lumenvox only)	SR audio 2005 05 14 10.24.32.0161 port 001.gmr1
32	GrammarAdd	File name containing additional grammar text (Lumenvox only)	SR audio 2005 05 14 10.24.33.0662 port 001.gmr2
33	SpeechStartDate	Date speech detected	2005/03/09
34	SpeechStartTime	Time speech detected	57901.730
35	SpeechEndDate	Date end-of-speech detected	2005/03/09
36	SpeechEndTime	Time end-of-speech detected	57907.053
37	SRAudioFile	Audio file name where SR input buffer recorded (name only)	SR audio 2005 02 14 19.54.53.0573 port 001.pcm

38	SRHeard	Result "heard" by recognizer	zero two eight two five nine oh
39	SRAdjusted	Result adjusted by teleflow	0282590
40	SRResultConf	Confidence score for this result	74
41	SRTermDigit	Digit that terminated an SR input, either empty or single digit	2
42	SRReturn	TeleFlow return value (SUCCESS, FAILURE, TIMEOUT, etc)	SUCCESS
43	SRHandle	TeleFlow SR result handle value (integer)	18

**Notes:**

Fields **CallStartDate**, **CallStartTime**, and **Port** (1-3) form a unique call identifier. Data from multiple files (ie: multiple ports, but same day) should be merged and sorted by this unique identifiers, so SR data is reviewed in call order.

If the most recently played Play step (fields 5-12) played to completion, the most recently executed Play step (12-19) will be the same. If the most recently played Play step is interrupted, then the most recently executed Play step will be the last Play step processed by TFEngine before the SR step, the ExecResult will be "INTERRUPTED" and the ExecTo will be 0 ms (because the file never played).

Field **SRHeard** and **SRAdjusted** (38-39) may be the same. TeleFlow will only translate the string if needed.

For field **SRReturn** (42), TeleFlow return values translate to SRReturn values as follows:

SRReturn	Means	TeleFlow status
"SUCCESS"	Successful recognition	IVR_SUCCESS
"FAILURE"	Recognition failed due to insufficient SR resources	IVR_FAIL
"FATAL"	Fatal error occurred	IVR_FATAL
"TIMEOUT"	No voice (or TT) heard in the allotted time	IVR_TIMEOUT
"HANGUP"	Caller hung up	IVR_HANGUP
"THRESHHOLD"	Recognition confidence failure, or digit entered instead (confidence 0% and digit in field 34, SRTermDigit)	IVR_THRESHOLD
"STATUS_" + #	Indicates a status value that is not known by the SpeechTrainer. The number represents the actual TFEngine status code, i.e.: STATUS_12	{other, # = status code}

For field **SRHandle** (43), the handle is a 32-bit unsigned integer. This number IS NOT a unique identifier, as each port has its own incrementing handle. The number starts at 0 when TFServer starts and will continue to increment for every recognition processed by that port.

### 13.2.5 Summary File

Generated once per TFServer startup.

It contains information about the TFServer environment in use at the time.

**Location:**

TeleFlow log directory

**File name:**

ST{server\_start\_date}.CSV

**where:**

{server\_start\_date} = The date TFServer started in the format yyyyymmdd

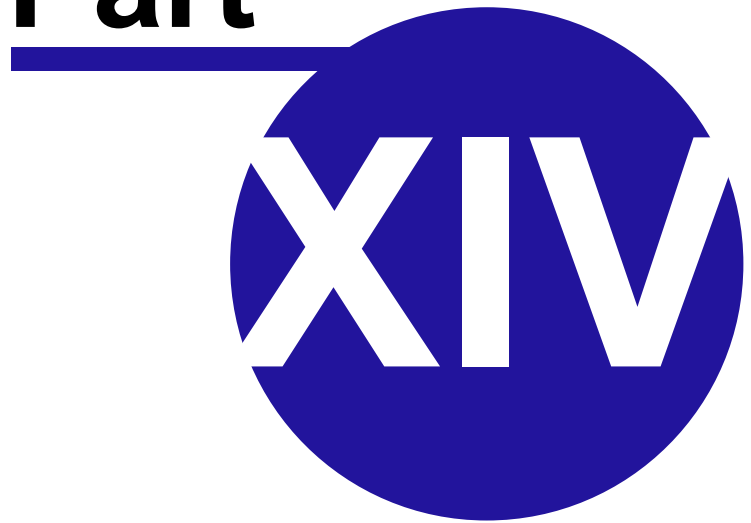
**Data:**

#	Field	Description	Example
1	TFStarteDate	Date TFServer started, in TF format	2005/03/09
2	TFStartTime	Time TFServer started, sec since midnight	153011
3	InstallDir	Location where TeleFlow installed	C:\Program Files\TeleFlow\
4	LogDir	Log directory	C:\myProject\Log\
5	MaxLogSize	Maximum size of log file, or -1 if unlimited	500
6	LogLines	Max number of characters logged on a FATAL condition	200
7	BoardDLL	Hardware DLL file	C:\Program Files\TeleFlow\brd_nms.dll
8	TALFile	Application line list file	C:\Program Files\TeleFlow\Default.tal



**TeleFlow**

**Part**



## 14 Appendix 2 - Third Party Links and Requirements

[Lumenvox Speech Recognition](#)

[Cool Edit](#)

[Microsoft Product Support](#)

[Nuance Speech Recognition](#)

### 14.1 Lumenvox Speech Recognition

To use speech recognition with [LumenVox 6.0](#), the SR Acquire step requires a LumenVox parameter file. LumenVox parameters are not case sensitive. Currently, only 3 options are supported in the LumenVox parameter file:

LogLevel: LumenVox log level parameter (0 to 6)  
Language: LumenVox phonym set (English, Spanish, or French)  
Tuner: Enable/disable LumenVox Speech Tuner (on or off)

i.e.

```
LogLevel 4  
Language English  
Tuner on
```

LumenVox grammars must be written according to the W3C Speech Recognition Grammar Specification (SRGS) Version 1.0. SRGS grammars can be used for base grammars or additional grammars in the SR Get String and Menu steps.

i.e.

```
#ABNF 1.0;  
language en-US;  
mode voice;  
root $yesno;  
$yes = (yes $GARBAGE) | yep | yeah | sure | certainly | ($GARBAGE (right | correct));  
$no = ((no | now | nah) $GARBAGE) | nope;  
$yesno = $yes{$=\\"yes\\";} | $no{$=\\"no\\"};
```

### 14.2 Cool Edit

Audio editor is now Adobe Audition

[Cool Edit](#)

### 14.3 Microsoft Product Support

Although enGenic is a built on windows, we do not support Microsoft products.

Please contact your Microsoft OEM dealer for support instructions or Microsoft directly using the link below.

[Microsoft Product Support](#)

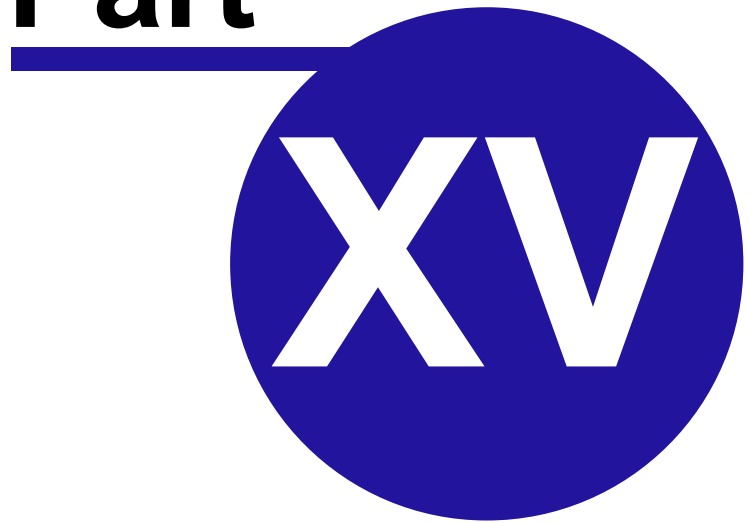
## 14.4 Nuance Speech Recognition

To use speech recognition with [Nuance 7.04](#), the SR Acquire step requires the directory of the Nuance pre-compiled grammars.

Nuance pre-compiled grammars can only be used as base grammars in the SR Get String and Menu steps. Additional grammars are not supported for Nuance.

**TeleFlow**

**Part**



## 15 Appendix 3 - Misc

[Database and ODBC Issues](#)

[ODBC Setup](#)

[SQL](#)

[Hardware Installation](#)

### 15.1 Database and ODBC Issues

[ODBC Setup](#)

[SQL](#)

#### 15.1.1 ODBC Setup

[Installing ODBC Drivers](#)

SQL Server Data Access Setup

Setting up and Connecting to an Oracle Database

##### 15.1.1.1 Setting up ODBC Data Sources

###### Installing ODBC Drivers

Installing Access, FoxPro, MySQL, SQLServer and other database products will automatically provide ODBC drivers. Once the appropriate drivers are available on the computer, data sources can be set up for applications. The method for the common data source setup - with some pointers for specific driver setup - is detailed below.

###### Configuring Data Sources

- Start >> Settings >> Control Panel >> ODBC.
- On the "User DSN" tab (the first page) click "Add".
- Choose driver to add:
  - For Microsoft Visual FoxPro: "\*.dbf"
  - For MS Access: "Microsoft Access Driver (\*.mdb)"
- Click "Finish".
- Driver Setup box will appear.
- Fill in the field "Data Source Name". This is the name that you will use from within TeleFlow applications to access your database. It is recommended that you choose a name that is unique and easy to remember. For example: "dBASEFile".
- Click 'OK' on the ODBC Driver Setup Box.
- Click 'OK' on the ODBC Data Source Administrator.

###### MS Access Example

- Locate and select the appropriate .mdb file.

When creating a TeleFlow application, the [Connect DB step](#) can be used to connect to the Data Source.

#### 15.1.2 SQL

[SQL User's Guide](#)

[SQL Notes](#)

### 15.1.2.1 SQL Notes

\*\*\* End each statement with a semi-colon

#### Group by Clause

Returns aggregate values by group. Eg: Select dept\_id, sum(Salary) from emp group by dept\_id; (This gives total of salaries for each department).

#### Having Clause

Specifies conditions for groups of records to display (Format: Having expr1 rel\_operator expr2)  
Eg/ Select Dept\_id, sum(salary) from emp group by dept\_id having sum(salary) > 200000; Make sure group by clause is used.

#### Union Operator

Union – Combines results of 2 select statements into a single result. Duplicate records are not returned.

Union All – Duplicate records are returned.

Statements must have an equal number of column expressions and same data types.

#### Order by Clause

Indicates how the records are to be sorted. Eg/ Select Emp\_id, last\_name, first\_name from emp order by last\_name;

#### For Update Clause

Locks records so no one can edit until the lock is removed.

COMMAND	DESCRIPTION
Desc	Describes database table (ie/ lists all fields)
Select	Get you information from selected fields Eg: Select name from mydb;
Update	Updates current database Eg/ Update \dbflavrssyst set sts_date = date() where status_cd = 'system';
Distinct	Filters out duplicate records

#### Aggregate Functions

Sum	Total of values in numeric field expression
Avg	Averages values in numeric field expression
Count	Returns number of values in any field expression
Max	Returns largest value in field expression

Min	Returns smallest value in field expression
From	Indicates tables that will be used in select statement
Where	Specifies conditions that must be met to be retrieved (Format:Where Expr1 rel_operator expr2) Eg: Select name from emp where salary 20000;

### 15.1.2.2 SQL User's Guide

TeleFlow has ODBC database access built in. Using a standard form of SQL, a wide range of data management tools and functions can be employed in any TeleFlow application. This guide is intended as a basic overview of SQL and its operations within TeleFlow. There are however, many books and web sites available to provide greater detail about SQL programming. Please also note that variations are required for different databases. As dBase files are common on desktop computers, they are used in the following examples.

#### SQL handles:

Each SQL statement must be assigned a number, or handle, in order for all subsequent statements to be controlled. This number must range from one to nine. For instance, several select statements can be defined. A select statement will not do anything until a fetch is encountered. The fetch knows which select statement to work from by the number of the handle. In this way, SQL steps can work with one another or independently on several open SQL statements at the same time.

SQL select statement format

```

SELECT [DISTINCT]
    { * | column_expression, ... }
FROM
    table_names [table_alias] ...
[ WHERE
    expr1 rel_operator expr2 ]
[ GROUP BY
    {column_expression, ...} ]
[ HAVING
    expr1 rel_operator expr2 ]
[ UNION
    [ALL] (SELECT...) ]
[ ORDER BY
    {sort_expression [DESC | ASC]}, ... ]

```

```
[ FOR UPDATE
  [ OF {column_expression, ...} ] ]
```

### SQL statements and TeleFlow variables

The definition of the SQL statement. TeleFlow allows a standard form of SQL to be used. e.g.: The following example shows what a select statement might look like. TeleFlow variables are populated in the 'select' section, and used to locate information in the where clause.

```
SELECT
  ACCNT1 + ACCNT2 + ACCNT3 ACCTOT,
  SHIPTO          SHIPTO,
  SHIPDATE       SHIPDATE
FROM
  \DATA\ACCOUNT.DBF
WHERE
  ACCTNO = @ACCTNO
```

Important: Note that ACCTOT, SHIPTO and SHIPDATE are defined at the end of each select line. TeleFlow will create these items as variables. In this regard, ACCTOT become @ACCTOT, SHIPTO becomes @SHIPTO and SHIPDATE becomes @SHIPDATE. Consider the following example:

```
SELECT
  SHIPTO,
  SHIPDATE
FROM
  \DATA\ACCOUNT.DBF
WHERE
  ACCTNO = @ACCTNO
```

In this case, SHIPTO and SHIPDATE are created as @SHIPTO and @SHIPDATE as there is no ambiguous functions to include with the naming of the variables.

```
SELECT
  MAX(TOTALSALE) MAXSALE
FROM
  \DATA\ACCOUNT.DBF
```

In this example, the select statement will find the largest sale value in the table and store it in the TeleFlow variable @MAXSALE.

### SQL aggregate functions

The following aggregate functions are available for use within SQL.

SUM	e.g.: SUM(SALARY)
AVG	e.g.: AVG(SALARY)
COUNT	e.g.: COUNT(NAME) or COUNT(*)
MAX	e.g.: MAX(SALARY)
MIN	e.g.: MIN(SALARY)

### SQL numeric operators



The following numeric operators are available for use within SQL.

- + Addition
- Subtraction
- \* Multiplication
- / Division
- \*\* Exponentiation
- ^ Exponentiation

### SQL character operators

The following character operators are available for use within SQL.

- + Concatenation keeping trailing blanks
- Concatenation moving trailing blanks to the end

### SQL date operators

The following date operators are available for use within SQL. Please note that SQL dates are often not compatible with TeleFlow dates. This is due to the way they are stored within different database systems. The following operators can be used to perform date math directly on the data as it is gathered from the database. Once the final date has been generated by functions and operators, it will need to be formatted to YYYY/MM/DD before it will be a valid TeleFlow date.

- + Add a number of days
- Subtract a number of days

### SQL relational operators

The following relational operators will be located in the where section of SQL statements

=	Equal
<>	Not equal
>	Greater than
<	Less than
>=	Greater than or equal
<	Less than
<=	Less than or equal
Like	Matching a pattern e.g.: where name like 'JOHN%' (note: % means match anything)
Not Like	Not matching a pattern
Is Null	Equal to null

Is Not Null	Not equal to null
Between	Range of values between a lower and upper bound
In	A member of a set specified of specified values or a member of a subquery (i.e. another SQL statement)
Exists	True if a subquery returned at least one record
Any	Compares a value to each value returned by a subquery. Any must be prefaced by =, <>, >, >=, <, or <=. =Any is equivalent to In
All	Compares a value to each value returned by a subquery. All must be prefaced by =, <>, >, >= or <= .

## SQL functions

The following functions are available for use within SQL

Functions that return character strings

- CHR
- RTRIM
- TRIM
- LTRIM
- UPPER
- LOWER
- LEFT
- RIGHT
- SUBSTR
- SPACE
- DTOC
- DTOS
- IFF
- STR
- STRVAL
- TIME
- USERNAME

Functions that return numbers

- MOD
- LEN
- MONTH
- DAY
- YEAR
- MAX
- DAYOFWEEK
- MIN
- POW
- INT
- ROUND

- NUMVAL
- VAL

Functions that return dates

- DATE
- TODAY
- DATEVAL
- CTOD

## 15.2 Hardware Installation

Consult the installation instructions provided by your hardware vendor.

[Dialogic](#)  
[Natural Microsystems](#)

### 15.2.1 Dialogic

[Dialogic](#) voice processing boards

### 15.2.2 Natural Microsystems

[Natural Microsystems](#) voice processing boards

#### 15.2.2.1 NMS Conference Configuration

The following instructions assume that you have an NMS system fully configured without conferencing, and want to add conferencing. Natural Access versions supported are 2003-1 and older.

1. Add conferencing configuration to your board configuration file. You need to allocate DSP resources to at least one conference manager and at least one conference slave per manager.

- For AG boards, your board configuration file will typically be in C:\NMS\AG\CFG\. The following example shows how to add one conference manager with two conference slaves:

```
#-----
# NaturalConference:
# one group of three DSPs is loaded with NaturalConference
# to form one conferencing resource containing three DSPs
#-----
DSP.C5x[13].Image = agcnfm.c54    # conference manager
DSP.C5x[14..15].Image = agcnfs.c54 # conference slaves
```

- For CG boards, your board configuration file will typically be in C:\NMS\CG\CFG\. The following example shows how to configure the card for a conference group of 7 DSPs with one manager and six slaves:

```
#-----
# Seven DSPs are loaded with NaturalConference to
# form one conferencing resource containing 7 DSPs
#-----
DSP.C5x[2..8].Libs[1] = cg6mslib
```

```
DSP.C5x[2].Files = cgcnmf      # conference manager
DSP.C5x[3..8].Files = cgcnfs   # conference slaves
```

```
ConferencingStream.Enable = YES
```

2. Add conferencing service manager to your CTA.CFG file, usually found at C:\NMS\CTAccess\CFG\

```
#-----
# CT Access System Configuration (ctasys)
#
# Valid options are:
# Service = name, dll - tells the daemon about available "services"
#           - tells the CT Access server what "services"
#           to export
#
# Note: NCC should always precede ADI when both services
# are listed.
#-----
[ctasys]
Service = ncc, adimgr
Service = adi, adimgr
.
.
Service = cnf, cnfmgr
```

3. Create (or update) the Conference Configuration file, CNF.CFG, also at C:\NMS\CTAccess\CFG\. For the DSP command, specified the full range of DSPs reserved for conferencing, including the conference master.

```
#-----
#           cnf.cfg
#
# --- Natural Conference Configuration File ---
#-----

[CONFERENCING]

# Flags = NO_ECHO_CANCEL | NO_DTMF_CLAMPING
Flags = NO_COACHING

[RESOURCE 0]

Board = 0
DSP = 13..15 # total DSPs reserved in this conference group
```

4. Set TeleFlow to use the NMS conference board DLL:

- a. In TFConfig, set the Telephony hardware to "NMS Communications".
- b. Run regedit
- c. Go to key [HKEY\_LOCAL\_MACHINE\SOFTWARE\enGenic\TeleFlow Server]
- d. Modify value NMS COMMUNICATIONS, changing it from "brd\_nms.dll" to "brd\_nms\_conf.dll"

5. Restart the computer

(note: you could shutdown TFServer, restart the CTDaemon service, reboot the boards with OAMSYS, and restart TFServer. Rebooting the computer is easier, and avoids errors).

One conference resource with three DSPs allows roughly 30 conferees.

Once conference resource with seven DSPs should support up to 64 conferees.

Actual performance depends upon the options set in your Natural Conference configuration file (cnf.cfg), and the requirements of your application.

To calculate the DSP resources needed for an application, refer to the spreadsheet Conference\_Capacity\_Costing\_rev2.xls.

### 15.3 Best Practices for T1/E1 Cabling Management

Where IP networks can be unplugged and plugged back in at any time, T1/E1 circuits are very different. When a T1/E1 cable is disconnected from the PSTN, it will go into "alarm" after a few seconds or minutes. Once a circuit goes into alarm, the telecom provider will need to be contacted to reset the trunk before T1/E1 service will resume. During this time, all communications on this line will ring busy from the callers end.

For this reason, it is very important to label the cables and connections so that they are not assumed to be a network cable.

"DO NOT unplug this cable without replacing with a "loop-back" connector, or the circuit will go into alarm"

When removing cables, a T1/E1 circuit can be saved from going into alarm by replacing the cable with a "loop-back" connector when removing it from the demarcation jack. This provides the telecom provider with feedback at all times so that the "alarm" signal is not fired. Once the cables are ready to plug back resume service, simply remove the "loop-back" connector with the cable.

In case the circuit goes into alarm, have the number of the telecom provider's operations center (sometimes referred to as the "NOC") for service resets. This information should be posted by the T1/E1 demarcation panel.

Example:

Telecom Provider:  
ABC Telecom  
Operations Center: 818.555.1234

Circuit information  
Circuit Identification: FJ#348583  
Pilot number: 818.555.9999

# Index

- . -

.TAM 132

- @ -

@CALL\_ELAPSE 169

@CALL\_END 169

@CALL\_START 169

@DATE 171

@FATAL 172

@FORMFEED 172

@HANGUP 172

@LINEFEED 173

@PROMPT 174

@SPELLTONES 174

@SYS\_DATA 176

@SYSDIR 176

@SYSDRIVE 176

@TIME 176

@TIMEOUT 177

@VOLUME 178

- A -

Action Steps 45, 46, 47, 48, 49

Action Steps: Invalid 46

Action Steps: Unrecognized Voice 48

Action Steps: Yes/No 49

Action Steps: Busy 45

Action Steps: Failure 46

Action Steps: Telephone Buttons 47

Action Steps: Time-Out 48

Add to Fax Icon 55

Advanced Steps 139

Answer 60

Application Environment Settings 28

array 166

arrays 166

Audio Step 75

Automatically Loading Application in Simulator 239

- B -

BAND 116

barge in 139

Basic Script 131

BasicScript 13

Beep Icon 75

Begin Trans Icon 85

Best Practice 277

Block Port Icon 50

BNOT 116

BOR 116

Bridge a Call 228

BXOR 116

- C -

Call DLL 102

Called ID (DNIS) 235

Case 97

change hardware parameter 139

CHAR 117

Chat Line 228

Clear TT Icon 70

Clearing Touch Tones 70

close 217

close conference 53

Comment Icon 95

Comments 96

Compare Variable to Value Icon 98

CONCAT 120

Conference 50

conference handle 50

Conferencing 50

configure 200

configure application 198

Configure TeleFlow 198

Connect Point 94

Connect Point Icon 38

connection 211

Connector Icon 45

console 206

Copy File Icon 99

Create Directory 100

Create Folder Icon 100

create system health 216

Creating A New Application 16

## - D -

Data Step Icons 83, 84, 85, 86, 87, 88  
Data Step Icons: SQL End Icon 86  
Data Step Icons: Begin Trans Icon 85  
Data Step Icons: DB Connect Icon 83  
Data Step Icons: DB Disconnect Icon 84  
Data Step Icons: End Trans Icon 86  
Data Step Icons: Fetch Icon 87  
Data Step Icons: Lock Icon 84  
Data Step Icons: SQL Statement Icon 88  
DATA XML Steps 82  
Databases: Connecting to 83  
Databases: Disconnecting from 84  
Date Info Icon 100  
DB Connect Icon 83  
DB Disconnect Icon 84  
Decrypt 101  
Delete File Icon 102  
delimited list 166  
deploy 204, 206  
Deploy TeleFlow Application 228  
Designer 13, 21  
develop and test application 8  
develop application 8  
Development 198  
Diable Reorder 54  
Dialogic 239  
Dictation Step 228  
display 209  
DLL 102  
DLL Function 102  
DO I WHILE..... LOOP 120  
Doc Steps 94  
DTMF clamping 50

## - E -

E1 PBX Integration with AI-Logix 228  
Echo cancellation 50  
Edit Menu 181  
email 155  
Email Name With @ 235  
Enable Reorder 54  
Encrypt 104

End Trans Icon 86  
English to French 188  
English to Frnch Standard Word Set 188  
English Voice Prompts 183  
Environment options 28  
Error Notification 228  
Exisiting Flowchart 19  
Existing applications 18  
exit 122, 217  
External Program 133

## - F -

Failure 50  
FAQ's 13  
Fatal Error 58, 107  
Fatal Stop Icon 106  
Fax Clear Icon 55  
Fax Receive Icon 56  
Fax Send Icon 57  
Fax Step Icons 55, 56, 57  
Fax Step Icons: Fax Receive Icon 56  
Fax Step Icons: Fax Send Icon 57  
Fax Step Icons: Add to Fax Icon 55  
Fax Step Icons: Fax Clear Icon 55  
Fax Steps 55  
Faxing: Adding to the Fax Queue 55  
Faxing: Clearing the Fax Queue 55  
Faxing: Receiving a Fax 56  
Faxing: Sending a Fax 57  
Fetch Icon 87  
File Copy Function 235  
File Menu 181  
Financial Calc 106  
Finding a File 235  
Finish Step 44  
First Flowchart 20  
Flagged Results 255  
Flash Hook Icon 57  
Flowchart 20  
Flowchart Menus 22  
Flowchart Menus: File Menu 22  
Flowcharts 13, 18, 19, 20  
Flowcharts: Adding a New 18  
Flowcharts: Adding an Existing 19  
Flowcharts: Adding Step Icons to 20  
Flowcharts: Adding to an Application 18  
FOR 122

**- G -**

Generate Touch Tones 63, 82  
 Get Letter Icon 70  
 Get Multiple Touch Tones 71  
 Get One Touch Tone 72, 141  
 Get Pending Touch Tones 74  
 Get Pending TT 74  
 Get TT Icon 72, 141  
 Get TT String 235  
 Get TT String Icon 71  
 Get TT String Playback 235  
 Getting Around in TeleFlow Designer 13  
 Getting started 2, 13, 15, 16  
 Global Events 58  
 Go To Label Icon 108  
 GOSUB 123  
 GOTO 124

**- H -**

Hang Up Icon 59  
 Hang Up Supervision 228  
 hardware setting 139  
 Hash 108  
 hashing 108  
 health 213  
 Help Menu 181  
 Hints 35  
 Hints, Tips, and Tricks 13  
 Hot Menu 33  
 Hot Menus 33, 34  
 Hot Menu: Selected Area Hot Menu 34  
 Hot Menu: STEP OPTIONS Hot Menu 33  
 Hot Menu: TABLET OPTIONS Hot Menu 34  
 HTTP 154  
 HTTP Step 154

**- I -**

IF 125  
 implementing application 8  
 In/Out Step Icons 70, 71, 72, 74, 141  
 In/Out Step Icons: Get Letter Icon 70  
 In/Out Step Icons: Get TT Icon 72, 141  
 In/Out Step Icons: Clear TT Icon 70

In/Out Step Icons: Get TT String Icon 71  
 In/Out Step Icons: Pending TT Icon 74  
 Initiating Application By ANI 235  
 Input steps 69  
 install 4  
 installation 4  
 Interface 13, 21  
 Internet WWW Access 154  
 Interruptible Play File Skips Next Play on Touch Tone 235

**- J -**

Join a conference 51  
 join conference 51

**- L -**

Label Icon 109  
 Labels 37  
 Language 136  
 Language Support 228  
 Leave conference 53  
 LEFT 126  
 LEN 126  
 License Procedure 11  
 License Requirements 11  
 Licensing 8, 11  
 Linelist 198  
 Load Text File 109  
 Load Text Icon 109  
 Load Text Step 109  
 Loading Applications in Simulator 239  
 Lock Icon 84  
 Log 95, 134  
 Lowercase 127  
 lumenvox 266

**- M -**

Map 27  
 Maximum conference seats 50  
 Menu Management 35  
 Menu Management: Isolating a Repetitive Series of Icons into a Separate Flowchart 35  
 menu options 211  
 Mixed Boards 228



MOD 127  
 Modulus 98  
 Modulus 10 98  
 monitor 209  
 monitor application 8  
 monitor menu 211  
 MRCP 228

## - N -

New Application 16  
 New Flowchart 18  
 NEXT (STEP) 122  
 Nortel DMS100 pbx 228

## - O -

ODBC: Connecting to an ODBC Data Source 83  
 ODBC: Disconnecting from 84  
 Open Conference 50  
 Open Text File 109  
 Opening an Existing Application 18  
 options 28  
 Outbound Calls 228  
 overview 8, 213

## - P -

parameter 139  
 parse 166  
 Parse Icon 112  
 Pending TT Icon 74  
 Phone Action Steps: Busy 45  
 Phone Action Steps: Telephone Buttons 47  
 Phone Step 49  
 Phone Step Icons 50, 57, 59, 60, 63, 66, 68, 82  
 Phone Step Icons: Flash Hook Icon 57  
 Phone Step Icons: Place Call Icon 60  
 Phone Step Icons: Wait for Call Icon 68  
 Phone Step Icons: Block Port Icon 50  
 Phone Step Icons: Hang Up Icon 59  
 Phone Step Icons: Send TT 63, 82  
 Phone Step Icons: Switch Disconnect Icon 66  
 Phone Step Icons: Switch Icon 66  
 Phone Step Icons: Unblock Port Icon 68  
 Phone: Answering 68  
 Phone: Hanging Up 59

Phone: Placing a Call 60  
 Phone: Sending Touch Tones 63, 82  
 Phone: Stopping/blocking incoming calls 50  
 Phone: Unblocking incoming calls 68  
 Pick Up 60  
 Pickup 60  
 Place Call Icon 60  
 Planning and Development 246  
 Play Icon 76  
 Play Variable 138  
 Port to Port Communication 235  
 POS 127  
 Print 128  
 Print Step 228  
 Productions Options 198  
 Prog Step Icons 58, 96, 98, 99, 100, 102, 106, 107, 108, 109, 111, 112, 113, 114, 115, 132, 133, 135, 136, 137, 138, 139  
 Prog Step Icons: Compare Variable to Value Icon 98  
 Prog Step Icons: Create Folder Icon 100  
 Prog Step Icons: Date Info Icon 100  
 Prog Step Icons: Go To Label Icon 108  
 Prog Step Icons: Label Icon 109  
 Prog Step Icons: Parse Icon 112  
 Prog Step Icons: Reload Off Icon 114  
 Prog Step Icons: Reload On Icon 115  
 Prog Step Icons: Scan Folder Icon 135  
 Prog Step Icons: Set Global Reference Icon 58, 107  
 Prog Step Icons: Set Variable Icon 136  
 Prog Step Icons: Variable Math Icon 111  
 Prog Step Icons: Wait Icon 138  
 Prog Step Icons: Copy File Icon 99  
 Prog Step Icons: Delete File Icon 102  
 Prog Step Icons: Fatal Stop Icon 106  
 Prog Step Icons: Load Text Icon 109  
 Prog Step Icons: Random Number Icon 113  
 Prog Step Icons: Run Flowchart Icon 132  
 Prog Step Icons: Run Program Icon 133  
 Prog Step Icons: Set Private Icon 137  
 Prog Step Icons: Write Text Icon 96, 139  
 Programming 97  
 Programming Steps 97

## - Q -

Quick Map 27

**- R -**

Random Number Icon 113  
 Read Registry 114  
 Read Text File 109  
 Recognition Flagged Results file 255  
 Recognition Results file 255  
 Record Icon 77  
 Recorder 64  
 Recorder Tone 64  
 Register Lumenvox 266  
 Register SIP URI 62  
 Register TeleFlow 201  
 Registration 228  
 Reload Off Icon 114  
 Reload On Icon 115  
 REM I 128  
 Reorder 54  
 Reports 219  
 Right 128  
 Round 129  
 run 204, 206  
 Run Basic Substr 131  
 Run Flowchart Icon 132  
 Run Program Icon 133  
 RUN TAP 133

**- S -**

safe 217  
 Save 95  
 Save Log 95, 134  
 SaveLog 95, 134  
 Say Icon 78  
 Scan Directory 135  
 Scan Folder Icon 135  
 Search Directory 135  
 Search for text 30  
 Search for text or steps 30  
 Search Steps 30  
 Select Case 129  
 send email 155  
 Send Media to VoIP 63  
 Send TT 82  
 Sending Media 63  
 server 204, 217

Server Set Up Dialogic 4  
 service 204  
 Set Global Reference Icon 58, 107  
 Set Language 136  
 Set Play Var Icon 80, 138  
 Set Play Variable 138  
 Set Variable Icon 136, 137  
 Set Volume Icon 80  
 Setting Icons Up So The Flow is Easy to Follow 37  
 Setting TeleFlow Variables 131  
 shut down 217  
 shutdown 217  
 Simulator 195  
 Simulator: Recording Error 228  
 Speak Text Icon 81, 143  
 Speech Rec 140, 246  
 Speech Recognition 246  
 Speech Recognition 140, 152, 266  
 Speech Recognition information 255  
 Speech Recognition Voice Passwords 152  
 SpeechTrainer 255  
 SQL 271  
 SQL End Icon 86  
 SQL Statement Icon 88  
 SQL User's Guide 271  
 SQL: Fetch Rows 228  
 SQL: Stored Procedures 228  
 SR 246, 266  
 SR Barge-in Off Icon 145  
 SR Barge-in On Icon 145  
 SR Clear Voice Icon 146  
 SR Get Digits Icon 147  
 SR Get Word Icon 148  
 SR Get Yes/No Icon 150  
 SR Speck Text 143  
 SR VoicePass Auth 151  
 SR VoicePass Reg 152  
 Standard math operators 131  
 start 204, 206  
 Start and Finish Step 43  
 Start step 43  
 Starting point 43  
 Statistics 219  
 Statistics Reporter 219  
 Stats and Reports 13  
 status 213  
 Step 38  
 Step Comments 96

- Step Definitions and Usage 13
- Step Icons 45, 95
- Step Icons: Beep Icon 75
- Step Icons: Comment Icon 95
- Step Icons: Compare Variable to Value Icon 98
- Step Icons: Connector Icon 45
- Step Icons: Copy File Icon 99
- Step Icons: Create Folder Icon 100
- Step Icons: Date Info Icon 100
- Step Icons: Flash Hook Icon 57
- Step Icons: Get Letter Icon 70
- Step Icons: Get TT Icon 72, 141
- Step Icons: Go To Label Icon 108
- Step Icons: Load Text Icon 109
- Step Icons: Parse Icon 112
- Step Icons: Place Call Icon 60
- Step Icons: Random Number Icon 113
- Step Icons: Reload Off Icon 114
- Step Icons: Reload On Icon 115
- Step Icons: Scan Folder Icons 135
- Step Icons: Set Global Reference Icon 58, 107
- Step Icons: Set Variable Icon 136
- Step Icons: Set Volume Icon 80
- Step Icons: SQL End Icon 86
- Step Icons: SR Clear Voice Icon 146
- Step Icons: SR Get Word Icon 148
- Step Icons: Variable Math Icon 111
- Step Icons: Wait for Call Icon 68
- Step Icons: Wait Icon 138
- Step Icons: Add to Fax Icon 55
- Step Icons: Begin Trans Icon 85
- Step Icons: Block Port Icon 50
- Step Icons: Clear TT Icon 70
- Step Icons: DB Connect Icon 83
- Step Icons: DB Disconnect Icon 84
- Step Icons: Delete File Icon 102
- Step Icons: End Trans Icon 86
- Step Icons: Fatal Stop Icon 106
- Step Icons: Fax Clear Icon 55
- Step Icons: Fax Receive Icon 56
- Step Icons: Fax Send Icon 57
- Step Icons: Fetch Icon 87
- Step Icons: Get TT String Icon 71
- Step Icons: Hang Up Icon 59
- Step Icons: Label Icon 109
- Step Icons: Lock Icon 84
- Step Icons: Pending TT Icon 74
- Step Icons: Play Icon 76
- Step Icons: Record Icon 77
- Step Icons: Run Flowchart Icon 132
- Step Icons: Run Program Icon 133
- Step Icons: Say Icon 78
- Step Icons: Send TT 63, 82
- Step Icons: Set Play Var Icon 80, 138
- Step Icons: Set Private Icon 137
- Step Icons: Speak Text Icon 81, 143
- Step Icons: SQL Statement Icon 88
- Step Icons: SR Barge-in Off Icon 145
- Step Icons: SR Barge-in On Icon 145
- Step Icons: SR Get Digits Icon 147
- Step Icons: SR Get Yes/No Icon 150
- Step Icons: Switch Disconnect Icon 66
- Step Icons: Switch Icon 66
- Step Icons: Unblock Port Icon 68
- Step Icons: VR Initialize Icon 144
- Step Icons: VR Release Icon 151
- Step Icons: Write Text Icon 96, 139
- Step: Definitions and Usage 38
- Steps 27, 30, 40
- stop 217
- Stopping TeleFlow Server 207
- String Replacement 228
- SUBSTR 131
- Summary File 255
- Switch Disconnect Icon 66
- Switch Icon 66
- Switch Ports Together 235
- System 2
- system health 213, 216
- system overview 213
- System Requirements 2, 13
- System Variables 15, 167
- System Variables: @CALL\_ELAPSE 169
- System Variables: @CALL\_END 169
- System Variables: @CALL\_START 169
- System Variables: @DATE 171
- System Variables: @FATAL 172
- System Variables: @HANGUP 172
- System Variables: @PROMPT 174
- System Variables: @SPELLTONES 174
- System Variables: @TIME 176
- System Variables: @TIMEOUT 177
- System Variables: @VOLUME 178
- System Variables: @FORMFEED 172
- System Variables: @LINEFEED 173
- System Variables: @SYS\_DATA 176

System Variables: @SYSDIR 176  
 System Variables: @SYSDRIVE 176  
 SYSTEMS VARIABLES 178

## - T -

T1/E1 Cable 277  
 TAL 198  
 TAM 132  
 TAP 133  
 TAP Screen 14  
 TCP 156, 157  
 TCP Connect 156  
 TCP Disconnect 157  
 TCP Recieve 157  
 TCP Send 158  
 TeleFlow 13  
 TeleFlow Designer 13, 21  
 TeleFlow Monitor 209  
 TeleFlow Register 201  
 TeleFlow Reports 219  
 TeleFlow Server 204  
 TeleFlow Simulator 195  
 TeleFlow SQL Reporter 224  
 TeleFlow Stats 219  
 TeleFlow Steps 40  
 TeleFlow Variables 15  
 test application 8  
 Text 96  
 Text Files: Reading from 109  
 Text Files: Writing to 96, 139  
 Text to Speech 81, 143, 228  
 TF Simulator 239  
 TFConfigure 200  
 TFMonitor 209  
 TFServer 204  
 TFSQLReporter 224  
 THEN 125  
 Third Party 11  
 Third Party Licenses 11  
 Tips 35  
 Tone clamping 50  
 Toolbox 27  
 Toolbox Window 27  
 Tools Menu 182  
 Transfer 67  
 Transfer call 67  
 Tricks 35

TRIM 132

## - U -

UDP Bind 159  
 UDP Close 160  
 UDP Recieve 160  
 UDP Send 161  
 Unblock Port Icon 68  
 UnRegister LumenVox 266  
 UPPERCASE 132  
 Using Labels 37  
 UTC 178  
 UTC Date 178  
 UTC Time 178

## - V -

Variable case sensitivity 235  
 Variable Math Icon 111  
 Variable Scope 15  
 Variable Usage 162  
 Variable Usage: Date Variables 164  
 Variable Usage: Real and Integer Variables 164  
 Variable Usage: Setting User Defined Variables in a Single Flowchart 162  
 Variable Usage: String Variables 164  
 Variables 13  
 Variables and Data Types 15  
 Voice Detection 228  
 Voice Passwords 152  
 Voice Prompts 13, 183  
 Voice Script Manager 13, 179, 181, 182  
 Voice Script Manager: Description 179  
 Voice Script Manager: File Menu 181  
 Voice Script Manager: Help Menu 181  
 Voice Script Manager: Tools Menu 182  
 Voice Script Manager: Windows Menu 182  
 Voice Script Manager: Edit Menu 181  
 Voice Scripts 36, 183  
 Voice Scripts Manager 181, 183  
 Voice Scripts Manager: Editing Voice Scripts 183  
 Voice Scripts Manager: Recording Voice Files with the Script Manager 182  
 Voice Scripts Manager: Using the Voice Script Manager to Add Voice Scripts to an Application 182

Voice Scripts: Recording Voice Files with the Script Manager 182  
Voice Scripts: Separating and Re-Using Repetitive Voice Scripts 36  
Voice Scripts: Using Existing or Pre-Recorded Voice Files 183  
Voice Scripts: Using the Voice Script Manager to Add Voice Scripts to an Application 182  
Voice Step Icons 75, 76, 77, 78, 80, 81, 138, 143, 144, 145, 146, 147, 148, 150, 151  
Voice Step Icons: Beep Icon 75  
Voice Step Icons: Set Volume Icon 80  
Voice Step Icons: Play Icon 76  
Voice Step Icons: Record Icon 77  
Voice Step Icons: Say Icon 78  
Voice Step Icons: Set Play Var Icon 80, 138  
Voice Step Icons: Speak Text Icon 81, 143  
Voice Step Icons: SR Barge-in Off Icon 145  
Voice Step Icons: SR Barge-in ON Icon 145  
Voice Step Icons: SR Clear Voice Icon 146  
Voice Step Icons: SR Get Digits Icon 147  
Voice Step Icons: SR Get Word Icon 148  
Voice Step Icons: SR Get Yes/No Icon 150  
Voice Step Icons: VR Initialize Icon 144  
Voice Step Icons: VR Release Icon 151  
Voice Tab 75  
VoicePass 152  
VoicePass Auth 151  
VoIP 63  
VR Initialize Icon 144  
VR Release Icon 151

## - W -

Wait for Call Icon 68  
Wait Icon 138  
Web Steps 154  
Website access 154  
WebSteps TOC 40, 154, 162  
What is TeleFlow Simulator 195  
What is the TeleFlow Designer 13  
Windows Menu 182  
Write File 110  
Write File Step 110  
Write Text 96  
Write Text Icon 96, 139

## - X -

XML 89, 90, 92, 93  
XML Fetch 90  
XML Load 92  
XML Send 89  
XML Steps 82  
XML Write 93

**engenic**