

**Merlin Programmer for Kids** is a simple program to help kids, ages 5 - 8 (ages prior to their moving on to more advanced languages such as [Scratch](#)), learn about the concept of sequential programming by allowing them to make Merlin or any of the other Microsoft Agent characters perform actions, move, speak, listen for and make sounds in a predetermined manner. They will learn about do loops, conditional program continuation based on time, key strokes, spoken words, and more! I wrote it in 2003 for my twins, Tommy and Betsy, ages 7 then (now 16!), and although they enjoyed it most for making the characters speak naughty words and make silly sounds, they learned more. I am sure your kids will have fun with it!  
It is a totally FREE program.

This help file is intended for you, the parents. You must explain how this program works to your child, in a manner appropriate for his/her age. Obviously, the ability to understand and work with some features in this program will depend on your child's age. Soon enough he/she will amaze you! In order to help you, several example scripts are included. Please read the sections on **Menu** and **Programming**. They are very short, but informative.

Merlin Programmer for Kids home: [SciSoft Company](#)  
My home page: [Yale University Ear Lab](#)

Merlin Programmer for Kids

Exit Character Record Audio Help About

**Actions**

Show me  Program

Character: All

**General Commands**

- Do
- Hide (character)
- Hide All
- Interval (seconds)
- Loop (number)
- Report (seconds)
- Sequential (True/False)
- Show (character)
- Show All

**Script**

File Play Cancel hide

```
interval 0
sequential true

merlin> charpos 400,400
play> charpos 100,100
max> charpos 200,200
peedy> charpos 300,300


merlin> waitfor time=.2 quiet
sound drumroll
merlin> speak Lets go nuts,
guys!
```

**Sounds**

Hear me  Program

**Sounds for All**

- applause
- blip
- carbrake
- ding
- drumroll
- explode



Betsy, Daddy and Tommy  
playing Merlin Programmer  
for Kids a few years ago.

# Installation

New!

After you download <[Merlin Programmer for Kids v.5.5.8.msi](#)> into a folder, double click on it and the Merlin program will be installed. A shortcut will be



placed on the desktop, and when you first double click on it additional required components will be installed and then the program will start. You do not have to go through the details below anymore, but I leave the info below so you understand what is installed.

Okay, just in case the installation doesn't go the way I intended, here is what to do after installing the MSI file. Go to the Merlin programmer directory in Program Files directory and double click on these files to install components

First run this program (this is also accessible from the Windows Start Menu)  
<[Merlin\\_Component\\_Installer.exe](#)>

If nothing happens when you double click on it, then go to the Merlin Programmer directory and delete the file <[merlin\\_comp.txt](#)>. If that file exists in the directory then the Component Installer will not run.

If the Component installer runs and ends successfully, Merlin should start automatically. If not then try running Merlin -- if it works then no need to continue below.

If the Component Installer didn't work then run these in succession. If Win7 gives a message box asking if the installation of any of the components failed then ignore that message.

- 1) <[MSagent.exe](#)>
- 2) <[spchapi.exe](#)>
- 3) <[tv\\_enua.exe](#)>
- 4) <[mscsrgpcl.exe](#)>

Try running Merlin and it should work.

In Win7 you may need to run in Win 7 Basic Window mode or set compatibility mode in the shortcut to Disable desktop composition – else a pink box around the characters may appear!

---

Old Information – no need to do this with current version

**Important !!!! ---- Merlin Programmer for Kids will not run without first installing the Microsoft Agent characters and associated software available free from the web.**

Before installation of **Merlin Programmer for Kids** the following software must be installed

- 1) Microsoft Agent Core Components, V.2,
- 2) Lernout & Hauspie® TruVoice Text-To-Speech Engine
- 3) Microsoft speech recognition engine – if you want to have the characters recognize voice commands.
- 4) Microsoft Agent characters,

These are freely available from the web. You can get all required files at Microsoft's web and if you are connected to the internet you can click on the links below to go to the pages.

The following items are posted at

<http://www.microsoft.com/products/msagent/downloads.htm>

1. Microsoft Agent Core Components, V.2 is a file called < MSagent.exe>. Save it to a temporary folder and double click on it to run and install.

**Microsoft Agent and localization support:**

You should verify that your system meets the minimal [system requirements](#) before downloading Microsoft Agent.

Download and install the core components of Microsoft Agent by clicking the link below. Note, you do not need to install these core components if you are using one of the Microsoft Windows XP, Windows 2000 or Windows Me operating systems. The Microsoft Agent core components are already installed on your computer.

 [Download the Microsoft Agent core components \(395 KB exe\)](#)

Localized language components are libraries (DLL files) that add support for the dialogs, windows, tooltips, and balloon text of the core components in an additional language to the default English support. To add a language to the core components, select it from the list below and click the download link. Note, that this localization support alone will *not* make Microsoft Agent characters speak audibly in these languages via text-to-speech - to do so, a [text-to-speech engine](#) would also be required.

 [Download selected language component](#)

2. Lernout & Hauspie® TruVoice Text-To-Speech Engine is <tv\_enua.exe> for American English, but differs for other languages. Save it to a temporary folder and double click on it to run and install.

### **Text-to-speech engines:**

Text-to-speech engines provide speech **output** capabilities for Microsoft Agent and the Agent Character Editor. In the list below, select one of the available text-to-speech engines, then click the download link. Note that you must also install the [Agent language component](#) for the specified language, except for American and British English.

 [Download selected engine](#)

Can't find a language? You may want to check out [other third-party speech engines](#) that support Microsoft Agent. They may have the language or voice that you're looking for.

3. Microsoft's speech recognition engine. Save it to a temporary folder and double click on it to run and install.

### **Speech recognition engines:**

Speech recognition engines provide speech **input** capabilities for Microsoft Agent. At this time Microsoft provides a single U.S. English speech recognition engine for use with Microsoft Agent. To use this engine you should have a Pentium 120-MHz PC (or faster). Click the link below to download this speech engine.

 [Download the Microsoft® Speech Recognition Engine \(6 MB exe\)](#)

4. MS agent characters. Save it to a temporary folder and double click on it to run and install.

Goto <http://www.msagentring.org> and go to the character gallery.

On this page you can get a variety of characters. The examples in Merlin Programmer for Kids use the characters Merlin, Plany, Max, and Peedy. Be sure to get at least those. Follow the instructions on the web pages.

Characters have the <.acs> extension but are downloaded as an<exe> installation file which you should run to automatically put the character in the default directory, usually <C:/windows/msagent/chars>.

Another page with characters is <http://www.agentry.net>

The general page for MS Agent character information is <http://www.microsoft.com/products/msagent>

If the **Microsoft Agent** characters on your computer running the Windows XP operating system have stopped talking to you, you may need to install/re-install the [SAPI 4.0 speech runtime support](#).

## Installing Merlin Programmer for Kids

The installation program is supplied as a self-extracting file <merlin\_instal.exe>. It has been checked for viruses so it is safe to run. Place the file in a temporary directory and run it by double clicking. A few files will be extracted.

The files that are extracted are

installation.html -- this file is the one you are reading now!  
setup.exe

setup.ini  
InstMsiW.exe  
InstMsiA.exe  
MerlProg.msi -- this file is the Merlin msi file

After extracting the files, double click on **setup.exe** and the **Merlin Programmer for Kids** will be installed. An icon will be placed on the desktop. Double click on it and the program will start. Links to the program and help file are also in the Start Menu. For subsequent installations, upgrades or repairs you only need to double click on the msi file (**MerlProg.msi**). The default directory is “C:\Program Files\Merlin Programmer for Kids\”. Script files are in “C:\Program Files\Merlin Programmer for Kids\scripts\”, and sound (wav) files are in “C:\Program Files\Merlin Programmer for Kids\wav\”.

When upgrades become available, you should uninstall the program with the Merlin msi file (**MerlProg.msi**), then install with the new Merlin msi file.



## Menu



**Exit button** closes the program after querying.

**Character** button opens up three items to choose from. They are Show all characters (sequentially introduced by Merlin and then hidden), Hide all characters and Hide all except Merlin. The last item will also show Merlin if all the Character is hidden. When the program starts all characters on your computer become available for use.

**Record audio** button opens up a small window that controls the recording of audio if you have a sound card and microphone in your computer. Type a name in, hit record button, talk or make funny noises, then click save. The filename will appear in the **Sounds Window**, available for playback programming within the **Script Window** (see Programming section).



**Help** button shows this help file.

**About** button shows Program information, including registration status and version number. It also shows my email address.

## Programming Merlin

**Merlin Programmer for Kids** is a very simple program that will help young kids understand simple programming tasks. But hopefully the experience will prepare your kids for more challenging programming (and thinking) tasks.



The heart of the program is the **Script** window, where a sequential series of action commands are placed in the text box. Commands may be typed in, each on a single line, or they can be placed automatically by selecting and double clicking on items from the **Merlin Actions** window or **Sounds** window (see below). When the **Play** button is clicked, the script will be run. The **Cancel** button will cancel the script. The **Clear** button will erase everything in the Script window. Each action command will be highlighted in the **Script** window as the character acts out the command. A clock in the upper right of the main window will indicate the passage of time during the

execution of the script program. If the **hide** box is checked, then all windows are hidden during the execution of the script. The file button will open the **Load/Save Script** window where scripts can be saved or loaded to or from a disk file. Click on a file name or type one in the top text box. Then press **load** or **save**. **Load** puts the script into the **Script** window text area. Double clicking on a file name does the same. **Save** writes the Script text to a file on disk. After you write a script, save it for future use. The default directory is "C:\Program Files\Merlin Programmer for Kids\scripts\ ". The default script extension is ".csp". They are ASCII text files and can be edited with programs such as Notepad, or Wordpad, or other text editors.



The **Merlin Actions** window shows all the actions that the loaded character can perform. Not all characters have the same commands, so a script file that works with one character may produce errors with another character. The commands for each character can be displayed by selecting the characters name from the drop down box. General commands which affect all characters are also listed under “**All**”. If the **Show me** item is selected, then double clicking on any action command will cause the character to perform the action and speak the word. In this manner, your child can see what the character can do. If the **Program** item is selected, double clicking on an action command will insert that command into the **Script** Window. Most of the commands are standard, single word, Microsoft Agent character commands. However, some commands require special action, and are detailed below.

.



## Special Commands

Most command are preceded with the name of the character followed by ">" and a space, for example, "merlin> blink". Below are some special commands that require extra information supplied by your child. Some are global (affecting all characters), so they are not prefaced with the character's name. Type case (lower or upper) does not matter.

**Pause (seconds)** --- "(seconds)" should be replaced by a number. This is a pause before continuing program execution.

**Interval (seconds)** --- "(seconds)" should be replaced by a number. This is the time waited between action commands. Since some commands take a long time to complete, making the interval too short will interfere with subsequent commands. The default interval is 5 seconds between commands.

**Sequential (true/false)** --- "(true/false)" is replaced by true or false. The default is true. If true, then the commands are followed sequentially, one by one down the list of script lines. If false the commands are executed as fast as possible so that characters may be able to perform simultaneous actions. Look at the "lets\_go\_nuts.csp" script. There are three sequential statements in that script, two of them are "true", and one is "false". Try changing the false to true, and see that now each character does his do loop seperatly in the order of the script. Change it back to false and they go nuts!

**Show (character)** --- "(character)" should be replaced by the name of a character, e.g. "Merlin", "Plany", etc. The names can be seen in the **Actions** window. Multiple characters can be shown simultaneously. In a script if you refer to a character in a command, for example, "plany> tickle 2" and Plany was not previously visible, it will be shown automatically.

**Hide (character)** --- "(character)" should be replaced by the name of a character, e.g. "Merlin", "Plany", etc. The names can

be seen in the **Actions** window.

**Hide all** --- Hides all characters.

**Character> Speak (words)** --- In the **Script** window, “(words)” should be replaced by any word or words or funny sounding letters! If you have created an audio file and placed it in the “wav” directory (this is done automatically when audio recording is done in Merlin Programmer for Kids), you can use this file for lip-syncing. Simply place the filename after the **Speak** command. For example, the following line will have Merlin lip-sync the speech in the wave file “my\_voice.wav”.

```
Merlin> Speak my_voice.wav
```

Look at the script file “silly voice.csp” for a functional example.

**Character> MoveTo X,Y** --- In the **Script** window, the “X ,Y” must be replaced by screen coordinates, for example, “132, 200”. At the bottom right of the main program window the X,Y coordinates of the mouse cursor and the character are displayed so that your child can drag the character (mouse



down on the character, press left button, and move) or move the mouse cursor to a particular location on the screen where he/she wants the character to move. He/she can then use those coordinates in the command.

**Character> MoveTo random** --- the computer will randomly choose the X,Y coordinates.

**Character> MoveTo (character)** --- In the **Script** window, the “(character) “ should be replaced by the name of a character, for example “merlin> moveto plany”. The first character will move to the second character’s position.

The **MoveTo** commands can have a speed parameter appended to them that will adjust the speed of the movements. The choices are **slowest**, **slower**, **slow**, **medium**, **fast**, **faster**, and **fastest**. For example, "**merlin> moveto random fastest**" would move Merlin to a random location very fast.

**Character> StopMoving** --- the character will stop moving ununconditionally..

**Character> CharPos (X,Y)** --- In the **Script** window, the "(X ,Y)" must be replaced by screen coordinates, for example, "132, 200". This command quickly sets the character to a new location on the screen. If the character is hidden the repositioning will occur without showing the character.

**Character> Tickle (size)** --- In the **Script** window, the "(size)" must be replaced by a number indicating how big a wiggle the character will make.

**Character> CharSize (W,H)** --- In the **Script** window, the "(W,H)" must be replaced by desired width, height of the character in pixels. The current and default settings are reported when this command is double clicked in the **Merlin Actions** window with the **Show me** item selected.

**Report (seconds)** --- In the **Script** window, the "(seconds)" should be replaced by a number of seconds. For those commands where the character counts down or waits for something to happen, the character typically repeats some phrase every number of seconds. This command should precede those commands to have an effect. The default is every 5 seconds.

**Character> Waitfor Time=(H:M AM/PM)** --- The "(H:M AM/PM)" should be replaced by a valid time, as shown at the bottom left in the information panel of the program. For

example, “2:31 AM” or “10:02 PM”. The character will continually say “I am waiting for....”, until the condition (time is reached) is met or exceeded. You can add the word “**quiet**” to the end of the command to stop the chatter, otherwise the character counts down every **Report** number of seconds (see **Report** command).

A screenshot of a system status bar with a blue border. The text inside reads "The time is 3:02 PM : Today is Friday, Jan 3 2003".

The time is 3:02 PM : Today is Friday, Jan 3 2003

**Character> Waitfor Time=(minutes)** --- In the **Script** window, the “(minutes)” should be replaced by a number of minutes, which can be fractional; for example “1.5”. The character will continually say “I am waiting for....”, until the condition (countdown to zero) is met or exceeded. You can add the word “**quiet**” to the end of the command to stop the chatter, otherwise the character counts down every **Report** number of seconds (see **Report** command).

**Character> Waitfor Key=(letter)** --- In the **Script** window, the “(letter)” should be replaced by the letter of a key on the keyboard. The character will continually say “I am waiting for....”, until the condition (pressing the key) is met or exceeded. You can add the word “**quiet**” to the end of the command to stop the chatter, otherwise the character counts down every **Report** number of seconds (see **Report** command). Double clicking on the character will cancel the wait and go to the next script statement.

**Character> Waitfor Key=anykey** --- The character will continually say “I am waiting for any key to be pressed.”, until the condition (pressing any key) is met. You can add the word “**quiet**” to the end of the command to stop the chatter, otherwise the character counts down every **Report** number of seconds (see **Report** command). Double clicking on the character will cancel the wait and go to the next script statement. You can test for the key pressed with the **If** statement (see below).



**Character> Waitfor Listen=anywords** --- The character will simply wait for something to be said. A small text box below the character will indicate this. When the condition is met, that is, one of the voice commands is spoken and understood by the character, execution will continue. The character thinks the words that it is waiting for every **Report** number of seconds (see **Report** command). When it recognizes a word or phrase then it speaks that word or phrase. You can add the word "**quiet**" to the end of the command to stop the repeat. Double clicking on the character will cancel the wait and go to the next script statement. You can test for the spoken word or words with the **If** statement (see below). Voice commands that will be checked for are introduced using the **If** statement. Speaking "cancel script" will cancel the script.

**Character> Waitfor Collide=(character)** --- In the **Script** window, the "(character)" should be replaced by the name of a character. The character will wait for the two characters to overlap each other on the screen before execution will continue. The character will continually say "I am waiting for....", until the condition (characters bump each other) is met or exceeded. You can add the word "**quiet**" to the end of the command to stop the chatter, otherwise the character counts down every **Report** number of seconds (see **Report** command). Double clicking on the character will cancel the wait and go to the next script statement. See the "collide.csp" script for a possible use of this command. In that script **Sequential** is set to false otherwise the **Waitfor Collide** command is not executed until after the characters stop moving. You would have drag the characters together to satisfy the overlap condition.

**Character> Waitfor Collide=anybody** --- The character will wait for any other character to overlap it on the screen before execution will continue. The character will continually say "I am waiting for....", until the condition (characters bump each other) is met or exceeded. You can add the word "**quiet**" to the

end of the command to stop the chatter, otherwise the character counts down every **Report** number of seconds (see **Report** command). You can test which characters bumped with the **If** statement (see below).

**Character> Waitfor GetOff=(character) ---** In the **Script** window, the “(character)” should be replaced by the name of a character. The character will wait until the two characters are not overlapping on the screen before execution will continue. The character will continually say “I am waiting for....”, until the condition (characters to get off each other) is met or exceeded. You can add the word “**quiet**” to the end of the command to stop the chatter, otherwise the character counts down every **Report** number of seconds (see **Report** command). Double clicking on the character will cancel the wait and go to the next script statement. See the “collide.csp” script for a possible use of this command..

**Do ..... Loop (number) ---** These two commands are always used together. First insert the **Do** command in the Script window, then insert other commands, such as “**Moveto random**”, and finally insert the **Loop (number)** command, where “(number)” is replaced by the number of times the commands within the loop will be executed. For example,

```
Do
  Moveto random
  Speak I moved here!
Loop 3
```

This script will repeat the two commands within the loop 3 times.

**If ... then, elseif ... then, else, endif ---** These commands are used to check whether something has happened and then depending on the what happened to perform some commands.

In the following example Merlin waits for a key press then depending on which key is pressed, the proper response is spoken. The commands between **else** and **endif** are executed if the key pressed was not a, b and c. You may put a **do loop** within the **If** statements. You may only have one **If .. endif** in a script.

There are three conditions that can be checked for with the **If** statement. They are

1. **Key input.** In the following example Merlin waits for a key and says something depending on which key was pressed.

```
Merlin> waitfor key=anykey
```

```
If key=a then
```

```
Merlin> speak the "a" key was pressed
```

```
Elseif key=b then
```

```
Merlin> speak the "b" key was pressed
```

```
Elseif key=c then
```

```
Merlin> speak the "c" key was pressed
```

```
Else
```

```
Merlin> speak some other key was pressed
```

```
Endif
```

2. **Character collisions.** In the following example Max and Peedy start jumping around on the screen and Merlin waits for either character to bump him; then depending on which character bumped him, the proper response is spoken.

```
interval 0  
sequential false
```

```
do
max> moveto random
loop 50

do
peedy> moveto random
loop 50

merlin> waitfor collide=anybody

interval 1
sequential true

if collision=max and merlin then
merlin> speak max you bumped me!
max> speak sorry

elseif collision=peedy and merlin then
merlin> speak peedy you bumped me!
peedy> speak sorry

endif
```

3. **Voice input.** In the following example Merlin waits for something to be said. The words that he will wait for are the ones checked for in the **If...endif** statement. When Merlin recognizes one of the voice commands, the **if** statement will be executed and the response dictates what will follow.

```
interval 0
merlin> charpos 500,500
show merlin

merlin> waitfor listen=anywords quiet

if listen=hello then
merlin> speak Hi to you, too!

elseif listen=goodbye then

merlin> speak Goodbye.
hide merlin

elseif listen=Where is Max? then
```

```
merlin> speak He is coming now!  
max> charpos 600,100  
max> speak Hello guys!  
max> speak please press the "g" key and I will get tickled!  
max> waitfor key=g  
max> tickle 3  
endif
```

See the scripts “if\_collision.csp” , “if\_listenup.csp” and “choose\_guys.csp” for functional examples

**Goto (Label)** --- “(Label)” should be replaced by a special identifier (must be *label1*, *label2*, *label3*...*label10*) on any line; then, for example, the command **Goto label1** will cause execution to jump to the line immediately following the label, **label1**. In the following script, Merlin waits for the spoken words goodbye or hello. If hello is spoken execution repeats because it jumps back to the line following **label1**, but if goodbye is spoken, execution eventually ends.

```
interval 1  
  
label1  
merlin> waitfor listen=anywords quiet  
  
if listen=goodbye then  
  
sound drumroll  
  
elseif listen=hello then  
  
sound applause  
goto label1  
  
endif  
  
sound blip
```

See “Goto.csp” for this functional script.



# Sounds

**Sound (wav file)** -- In the script window, the “(wav file)” should be replaced with one of the sounds (such as “blip”) in the **Sounds** Window. Your child can create these sounds by using the Sound recorder accessed through the **Menu** bar, or the sound files can be any standard Microsoft audio “wav” files that are



placed in the directory “C:\Program Files\Merlin Programmer for Kids\wav\ “.If the **Program** item is selected, the sound command can be automatically inserted into the **Script** window by double clicking on the sound name in the Sound Window. If the **Hear Me** item is selected then your child can listen to the sounds before using them in the script.

## Running the Script

Finally, your kids can see their creation outside the programming environment by creating a short cut on the desktop that runs **Merlin Programmer for Kids**. Right click on the desktop, select create new shortcut. Then simply browse to find a script file in “C:\Program Files\Merlin Programmer for Kids\scripts\ “. Select that file and finish the shortcut. When your child double clicks on that shortcut icon on the desktop, the script will run on the desktop in the apparent absence of the programming environment, and it will look like an independent program has been produced. The script will end automatically. You can also double click on the script file in Explorer to run it.



## Disk Space

All parents know that disk space is important. **Merlin Programmer for Kids** shows available disk space for all drives at the bottom right of the program in the information panel. If your child double clicks on the panel, then the free space on each drive of his/her computer will be displayed sequentially.



## **Exiting the Program**

In general, double clicking on the character will cancel script execution. Hitting the “esc” key will cancel the script. Hitting <control><q> will quit the program unconditionally. Right clicking on the character will open up a window and if you click on “Quit” the program will close unconditionally. You can also cancel a script from this window. Clicking on the character with the <shift> key down will terminate the program. All of these options to quit are to ensure that the program will not get stuck in the running state if you do not want it to continue running!