

# Speech Engine Support Requirements

---

## ActiveX™ Technology for Interactive Software Agents



September 1997  
Microsoft Corporation

**Note:** This document is provided for informational purposes only and Microsoft makes no warranties, either expressed or implied, in this document. The entire risk of the use or the results of this document remains with the user.

Information in this document is subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. The furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property rights. Microsoft, MS, MS-DOS, Windows, Windows NT, and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the U.S. and/or other countries. Other product and company names mentioned herein may be the trademarks of their respective owners.

## Contents

Introduction

Requirements for Text-To-Speech Engines

Requirements for Speech Recognition Engines

## Introduction

Microsoft Agent uses the Microsoft Speech Application Programming Interface (SAPI) to support speech input (speech recognition, or SR) and speech output (text-to-speech, or TTS). By supporting this standard, Microsoft Agent's speech services can be supported by other

speech engines. This document describes the required SAPI interfaces used by Microsoft Agent.

## Requirements for Text-To-Speech Engines

The engine must be fully SAPI 1.0-compliant. In addition, the engine must also support the following SAPI interfaces for tagged text and bookmark notifications. These interfaces enable Microsoft Agent to pace the output of text to a character's word balloon and lip-sync the character's mouth (or equivalent) with the spoken words.

### ITTSCentralW

The engine must support **TextData()**, **AudioReset()**, **Register()**, **Unregister()**, and **Inject()**.

### ITTSNotifySinkW

The engine must call out through **AudioStop()**, **AudioStart()**, and **Visual()**. The **Visual** callback must provide IPA phonemes. (The International Phonetic Alphabet [IPA] is a universal notation for describing the phonetic content of spoken communication. All speakable phonemes have representations in IPA. Details of IPA are in the Microsoft Speech API specification [part of the Speech SDK 3.0 download] at <http://research.microsoft.com/research/srg/install.htm>.)

Although the **Visual** notification is fairly rich, Microsoft Agent uses only the **cIPAPhoneme** value to animate the mouth as the character speaks. Any Microsoft Agent-compatible engine must provide a closely synchronized stream of **Visual** notifications reflecting the phonetic content of the produced utterance. In this case, "relatively timely notification" is not adequate, because speaker-hearers are fairly sensitive to discrepancies between mouth position and acoustic content. **Visual** notifications need to be returned promptly.

### ITTSBufNotifySinkW

The engine must call out through **BookMark()**. During preprocessing of speech output, Microsoft Agent code inserts bookmarks between "words" and uses the arrival of those bookmarks to drive the pacing of text in the word balloon. While SAPI does not require anything more than the arrival of those bookmarks at some time before the end of the utterance, the bookmarks must be returned in a relatively timely fashion to support Microsoft Agent.

Note that there is no strict concept of "word" in some languages, such as Japanese. Microsoft Agent's **Speak** method defines a "word" as a connected string of symbols that has a meaning and pronunciation in isolation. Microsoft Agent uses fairly simple parsing code to determine what a "word" is: it looks for symbols separated by white space. Thus, there are three "words" in the English string "The 101 Dalmatians": "the", "one hundred and one", and "dalmatians". (Text included in the Microsoft Agent Map tag is treated as a single "word" for display purposes.)

### ITTSAttributesW

The engine must support pitch and speed attributes through the **PitchSet()**, **PitchGet()**, **SpeedSet()**, and **SpeedGet()** methods.

## Requirements for Speech Recognition Engines

A speech recognition engine must also be a fully compliant Command and Control (C&C) engine according to SAPI 1.0. It must support multiple grammars in the binary format described in the specification and allow those grammars to be activated or deactivated in real time.

In addition, to be considered Microsoft Agent-compliant, the engine must return results objects upon the successful recognition of a phrase (through **ISRGramNotifySinkA::PhraseFinish**). These results objects must support **ISRResBasic**, as the specification requires. In addition, they should support **ISRResScore**, which is a SAPI 3.0 interface not found in SAPI 1.0. Although Microsoft Agent will run with an engine that supports only **ISRResBasic**, or even with an engine that returns no results objects whatsoever, performance will usually be significantly poorer with such engines. Many applications use the confidence values provided by the engine to control how they respond to various commands.