

W I N C A L I S
A U T H O R
U S E R ' S G U I D E

HUMANITIES COMPUTING LABORATORY
DURHAM, NORTH CAROLINA, USA

Copyright Information

© Copyright 2001 by the Humanities Computing Laboratory, Durham, North Carolina, USA.

All rights reserved.

No part of this publication may be reproduced, transmitted, or transcribed, stored into a retrieval system or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written consent of the Humanities Computing Laboratory.

The Humanities Computing Laboratory reserves the right to revise this publication and to make changes from time to time in the content of this publication without obligation of the Humanities Computing Laboratory to notify any person of such revision.

Microsoft is a registered trademark of Microsoft Corporation; Windows is a trademark of Microsoft Corporation.

IBM is a registered trademark of International Business Machines Corporation.

Unicode is a registered trademark of the Unicode Consortium.

Send your completed Registration Form, or direct any questions, suggestions, product orders, and comments to one of the following locations:

In the USA and non-Argentina:

Humanities Computing Laboratory

301 W. Main St., Ste. 400-I

Durham, NC 27701 USA

Phone: (919) 667-9556

Fax: (919) 667-9556

E-mail: info@humancomp.org

Web: <http://www.humancomp.org>

In Argentina:

Centro Blas Pascal

Investigación Y Desarrollo En Informática Educativa

Bacacay 705 - piso 11 "B"

Ciudad de Buenos Aires

Buenos Aires, Argentina

Voice: +54 1 4432-8353 / 8358

Fax: +54 1 4432-8353 / 8358 (15) 4181-9904

bpascal@ciudad.com.ar

Table of Contents

COPYRIGHT INFORMATION	II
TABLE OF CONTENTS	III
ABOUT THE WINCALIS AUTHOR USER'S MANUAL.....	XI
WHO YOU ARE AND WHERE TO GO	XI
<i>Beginning Users</i>	<i>xi</i>
<i>Intermediate and Advanced Users</i>	<i>xi</i>
SECTION SUMMARIES	XI
<i>Introduction: What is CALIS?</i>	<i>xi</i>
<i>Getting Started With WinCALIS</i>	<i>xi</i>
<i>Authoring Basics</i>	<i>xii</i>
<i>Lesson Development</i>	<i>xii</i>
<i>Customizing and Communicating</i>	<i>xii</i>
<i>The CALIScript Language</i>	<i>xii</i>
<i>Multimedia in WinCALIS</i>	<i>xii</i>
<i>Special Authoring and Editing Functions</i>	<i>xii</i>
<i>New Features in WinCALIS</i>	<i>xii</i>
<i>Specific Language Reference Manuals</i>	<i>xii</i>
<i>Configuring and Customizing WinCALIS Author</i>	<i>xii</i>
<i>WinCALIS Code and Command Conversions</i>	<i>xiii</i>
<i>CALIScript Language Quick Reference</i>	<i>xiii</i>
INTRODUCTION: WHAT IS CALIS?	1
THE STRENGTHS OF WINCALIS.....	1
<i>Structure</i>	2
<i>Note to Users of Text-Based CALIS</i>	2
HARDWARE AND SOFTWARE REQUIREMENTS	2
COPYRIGHT INFORMATION	3
WINCALIS REGISTRATION	3
WINCALIS SUPPORT SERVICES.....	4
<i>Electronic Mail</i>	4
<i>World Wide Web</i>	4
<i>Fax</i>	4
<i>Telephone</i>	5
<i>Address</i>	5
CHAPTER 1: GETTING STARTED WITH WINCALIS	7
SYSTEM REQUIREMENTS	7
INSTALLATION (WINCALIS).....	7
<i>Choose Optional Modules</i>	8
WINCALIS STRUCTURE	8
<i>Starting WinCALIS</i>	8
WINCALIS OPENING SCREEN MENUS AND LAYOUT	8
<i>Menu Bar</i>	8
<i>WinCALIS Opening Screen Layout</i>	10
WORKING A WINCALIS LESSON	11
<i>WinCALIS Menus</i>	11
<i>Accents and Diacritics</i>	17
CHAPTER 2: AUTHORING BASICS.....	19
BEFORE YOU BEGIN AUTHORING.....	19
YOUR FIRST SCRIPT	19
1. <i>Giving Directions</i>	20
2. <i>Creating a Presentation (Text) Window</i>	20
3. <i>Producing a Multiple Choice Question</i>	22

4. Producing a True/False, True Question and a True/False, False Question.....	24
5. Producing a Short Answer Question.....	24
6. Producing a Cloze Question.....	25
7. Producing a Tutorial Window.....	27
8. Saving the Lesson.....	29
Running a WinCALIS Lesson.....	29
WinCALIS Shortcut Keys	30
OVERVIEW OF WINCALIS AUTHOR.....	30
WINCALIS AUTHOR AND UNICODE.....	31
THE EDITOR.....	31
Control Menu.....	32
Title Bar.....	32
Maximize/Minimize Buttons.....	32
Restore Button.....	32
Vertical/Horizontal Scroll Bar.....	32
Window Border/Window Corner.....	32
Insertion Point.....	32
Mouse Pointer.....	32
WinCALIS Editor Limits.....	33
The WinCALIS Author Environment.....	34
WINCALIS AUTHOR WINDOWS.....	34
Lesson Description Window.....	34
Popup (Directions) Window.....	34
Presentation (Text) Window.....	35
Interactive (Question) Window (With Answers and Feedback).....	35
Scratchpad Window.....	35
Pass Through Window.....	36
WINCALIS AUTHOR MENUS.....	38
The File Menu.....	38
The Edit Menu.....	44
The L: [Language] Menu.....	48
The Format Menu.....	50
The Script Menu.....	52
The Task Menu.....	53
The Tools Menu.....	55
[Tools] Sort Lines.....	60
[Tools] Generate Index.....	60
[Tools] Set Tools.....	60
[Tools] Check Spelling.....	61
[Tools] Print File (TrueType).....	61
[Tools] Print File (Bitmap).....	61
[Tools] Windows Paintbrush.....	61
[Tools] Keyboard Macro.....	61
[Tools] Tools Menu Readme.....	61
The Window Menu.....	61
[Window] 1 Interactive (Question) (Ctrl+1).....	62
[Window] 2 Presentation (Text) (Ctrl+2).....	62
[Window] 3 Popup (Directions) (Ctrl+3).....	62
[Window] 4 Lesson Description (Ctrl+4).....	62
[Window] 5 Pass Through (Ctrl+5).....	62
[Window] 6 Scratchpad (Ctrl+6).....	63
Arranging Windows and Icons.....	63
To arrange windows on your desktop:.....	63
To arrange program-item icons:.....	63
To activate the editing window in WinCALIS Author:.....	63
The Help Menu.....	63
CHAPTER 3: LESSON DEVELOPMENT	65

OVERVIEW	65
POPUP (DIRECTIONS) WINDOW	65
INTERACTIVE (QUESTION) WINDOW	66
<i>Methodology</i>	66
<i>The Anticipated Response/Feedback Dialog Box</i>	66
WHAT'S YOUR QUESTION TYPE?	70
<i>Short Answer</i>	70
<i>Flash Card</i>	70
<i>Multiple Choice</i>	71
<i>True/False</i>	71
<i>Cloze Exercise</i>	72
MOVING AMONG TASKS WITH THE TASK MENU	76
<i>Next/Previous</i>	76
<i>Delete</i>	76
<i>Go to Task</i>	76
<i>Rearrange Tasks</i>	76
QUESTION SEQUENCING IN WINCALIS	76
GLOBAL (ENTIRE SCRIPT) AND TASK OPTIONS	77
<i>Check Case</i>	77
<i>Check Accents/Punctuation/Spelling</i>	78
<i>Mark Spelling</i>	78
<i>Checking the Score</i>	78
<i>All Answers</i>	78
<i>Allow Passing</i>	78
<i>Show Correct</i>	78
<i>Praise</i>	78
<i>Center Options</i>	78
<i>Giveup repeats task at end</i>	78
<i>Randomize Multiple Choice</i>	79
<i>Max. Tries</i>	79
<i>Min. Tries</i>	79
MORE OPTIONS	79
<i>Allow Repeat</i>	79
<i>Allow Giving Up</i>	80
<i>Pass Between Groups</i>	80
<i>All n Answers</i>	80
<i>Enter Number of Answers n:</i>	80
<i>Use D Window</i>	81
<i>Disable Keyboard</i>	81
<i>No Mouse Hand Pointer</i>	81
<i>Limit Answer Length</i>	81
<i>Enter Answer Length n:</i>	81
<i>Tab Stop in Multiple Choice:</i>	82
<i>Blank [Window]</i>	82
<i>Quiz</i>	82
<i>Interactive (Question) Window Top Line:</i>	82
<i>Left Margin Column Number:</i>	83
SCRIPT GENERATION OPTIONS	83
<i>Random Task Order</i>	83
<i>Student Chooses # of Tasks</i>	83
<i>Use [English/W.Europe] Keyboard</i>	83
<i>Info Line in Task Windows</i>	83
<i>Center Text in the Window</i>	83
RUNNING A LESSON IN WINCALIS	84
THE VALUE OF EXPERIMENTATION	84

CHAPTER 4: CUSTOMIZING AND COMMUNICATING.....	85
OVERVIEW	85
IT'S ALL IN THE WILD CARDS	85
<i>Definition</i>	85
<i>Implementation</i>	85
<i>The Wild Card Characters</i>	86
<i>Wild Cards Which Match Single Characters</i>	86
<i>Wild Cards Which Match Single Words</i>	86
<i>Wild Cards Which Match Anything</i>	87
<i>Using WinCALIS Wild Cards in Anticipated Responses</i>	87
<i>Going Further with Wild Cards</i>	89
<i>Spellmarking and Wild Card Use</i>	91
<i>Sequencing Answers and Wild Card Use</i>	91
<i>Correct Answer Sequencing</i>	91
<i>Anticipated Incorrect/Remain Here Answer Sequencing</i>	92
<i>Using Wild Card Characters as Literal Characters</i>	94
USING SPECIAL RESPONSES IN WINCALIS AUTHOR.....	94
<i>Using Special Responses: <BUTTON></i>	94
<i>Using Special Responses: <ENTER></i>	94
<i>Using Special Responses: <GIVEUP></i>	95
<i>Using Special Responses: <HOTSPOT></i>	96
<i>Using Special Responses: <IF></i>	96
<i>Using Special Responses: <KEY 'x'></i>	96
<i>Using Special Responses: <MOUSE: x y></i>	97
<i>Using Special Responses: <OR></i>	98
<i>Using Special Responses: <PASS></i>	99
<i>Using Special Responses: <PASSBACK></i>	100
AESTHETICS OF LESSON PRESENTATION	100
<i>Color Me CALIS</i>	100
<i>Setting Foreground and Background Colors</i>	101
<i>A Last Note on Color Use</i>	103
<i>The Bold, The Strikeout, and the Others</i>	103
<i>Positioning the Cursor and Text in Windows</i>	104
<i>Sizing Up Your Text</i>	105
<i>CENTERing a Text Line</i>	107
<i>Using TABs to Align Text On-Screen</i>	107
<i>TABbing</i>	108
<i>DELAYing Execution</i>	108
<i>Sounding a BEEP</i>	109
<i>Sizing Windows</i>	109
<i>Using the SIZE Command in WinCALIS Lessons</i>	110
<i>A Note on Commands</i>	111
<i>Refining Feedback</i>	111
<i>Displaying the LAST Student Entry and the CORRECT Answer</i>	111
<i>Addressing Windows when Giving Feedback</i>	113
<i>The Window Commands</i>	113
<i>MARK My Words: Positioning the Cursor at Specific MARKs in Your Text or Question</i>	116
<i>The Power of SPELLMARKing</i>	119
<i>Choosing the TOPLINE of your Windows</i>	125
<i>BLANKing and CLEARing Windows</i>	127
<i>Extra-CALIS Communication</i>	129
...AND MORE.....	134
CHAPTER 5: THE CALISCRIP LANGUAGE	135
ROLE OF CALISCRIP.....	135
RELATIONSHIP BETWEEN WINCALIS AND CALISCRIP	135
.CAL FILE STRUCTURE.....	136

<i>SAMPLE *.CAL SCRIPT</i>	136
READING AND WRITING *.CAL FILES	137
<i>To Load a *.CAL file into the Scratchpad:</i>	137
<i>To Translate a *.CAL file:</i>	137
<i>To import into the Scratchpad:</i>	138
CALIScript SYNTAX	138
<i>WinCALIS Statements</i>	138
<i>Statements and Statement Markers</i>	139
<i>General Rules for Statement Markers</i>	139
<i>The Popup (Directions) Window Statement</i>	143
<i>/ The Presentation (Text) Window Statement</i>	144
<i>? The Interactive (Question) Window Statement</i>	144
<i>Wild Cards</i>	147
WINCALIS COMMANDS	150
<i>Basic Presentation Commands</i>	151
WINDOW ACTIVATION COMMANDS.....	159
<i>Overview</i>	159
WINDOW MANIPULATION COMMANDS	164
<i>Media Commands</i>	178
<i>Sequential Flow of the Script through Task List Manipulation</i>	185
SPECIAL RESPONSES.....	196
TASK OPTIONS AND DEFAULTS	207
<i>Modifying the Appearance of Tasks</i>	209
<i>WinCALIS Answer Evaluation Options</i>	213
YOUR KEY TO CREATIVITY	221
CHAPTER 6: MULTIMEDIA IN WINCALIS	223
OVERVIEW OF MULTIMEDIA	223
<i>Multimedia Requirements</i>	223
<i>Function of Multimedia</i>	224
MORE TIPS ON MULTIMEDIA IN WINCALIS	224
<i>Background</i>	224
<i>Analog vs. Digital</i>	224
<i>MIDI and Audio CDs</i>	224
<i>Animation</i>	225
<i>Photographic Images</i>	225
DESIGNING A MULTIMEDIA LESSON	225
PREPARING MULTIMEDIA FILES.....	225
<i>Audio</i>	226
<i>Video</i>	226
<i>Animation</i>	226
RECORDING AUDIO	226
.WAV AND *.MMD FILES	227
<i>Creating *.MMD Files</i>	228
WINCALIS SUPPORT FOR MULTIMEDIA	228
<i>Audio</i>	228
<i>Video</i>	228
<i>Animation</i>	229
GRAPHICS.....	229
<i>Requirements</i>	229
<i>Creating Graphics</i>	229
<i>Saving Graphics</i>	229
<i>Graphics in WinCALIS Author</i>	229
<i>More Detailed Information on Graphics</i>	230
HOW TO INSERT GRAPHICS IN WINCALIS AUTHOR.....	232
WINCALIS MULTIMEDIA EDITOR.....	235

Start the Multimedia Editor	235
Multimedia Editor Menus	235
Selecting a Device	235
EDITING IN THE MULTIMEDIA EDITOR	236
<i>Multimedia Editor Controls:</i>	236
<i>Saving clips</i>	239
<i>To Edit an Existing *.MMD file</i>	239
<i>Saving the *.MMD Clip File</i>	240
<i>The Structure of the *.MMD Clip File</i>	240
<i>Editing the *.MMD File</i>	241
INSERTING MULTIMEDIA IN WINCALIS LESSONS	241
<i>Inserting Audio</i>	241
<i>Inserting Video or Animation</i>	242
<i>A Detailed Example of Inserting Video Clips</i>	243
AUDIO AND VIDEO OPTIONS	245
<i>Audio Options</i>	245
<i>Video and Animation Options</i>	246
<i>Size</i>	246
<i>Location</i>	247
<i>Play</i>	247
<i>Audio</i>	247
<i>Video</i>	247
<i>Zoom Image</i>	247
<i>MCI Control / Media Controls Bar</i>	247
SUMMARY OF AUDIO COMMANDS	248
<i>Audio Options and their Corresponding CALIS Commands</i>	248
SUMMARY OF VIDEO COMMANDS	248
<i>Video Options and their Corresponding CALIS Commands</i>	248
<i>Distributing Multimedia Exercises</i>	249
CHAPTER 7: SPECIAL AUTHORIZING AND EDITING FUNCTIONS	251
BROWSING AND INSERTING TEXT FROM UNICODE TABLES	251
<i>Unicode Character Selection Dialog Box</i>	251
PASTING TEXT WITH CODE CONVERSION	252
DICTIONARIES IN WINCALIS	254
<i>User-Supplied On-line Dictionaries</i>	254
JAPANESE-AND-ENGLISH DICTIONARY	258
<i>Key to Abbreviations</i>	258
KOREAN-AND-ENGLISH DICTIONARY	259
<i>Key to Abbreviations</i>	259
<i>Key to Symbols</i>	260
<i>General Features</i>	260
<i>Acknowledgments:</i>	262
HOW TO CREATE YOUR OWN ON-LINE DICTIONARY	262
<i>Create a dictionary text file</i>	263
<i>Generate the Dictionary Index</i>	263
<i>Associate with a Keyboard</i>	263
<i>Dictionary Creation Notes</i>	264
<i>Tools Menu Option "Generate Index"</i>	264
RECORD-KEEPING IN WINCALIS	265
WINCALIS RECORD-KEEPING: VIEWING THROUGH PARADOX FORMS	268
<i>Installation</i>	268
<i>Controls</i>	270
<i>Example</i>	271
<i>Editing Paradox Records</i>	271
<i>Previewing the Results</i>	272
<i>Deleting Files</i>	272
PRINTING A WINCALIS SCRIPT OR DOCUMENT	273

<i>WinCALIS Multilingual Printer Program (MLP.EXE)</i>	273
<i>WinCALIS TrueType Printer Program (MLPTT.EXE)</i>	274
CHAPTER 8: NEW FEATURES IN WINCALIS	277
<i>New WinCALIS Features and Extensions to CALIScript</i>	277
EVEN MORE NEW FEATURES IN WINCALIS:	289
<i>WinCALIS and WinCALIS Author:</i>	289
<i>WinCALIS:</i>	290
<i>WinCALIS Author:</i>	290
CHAPTER 9: SPECIFIC LANGUAGE REFERENCE MANUALS	291
MISCELLANEOUS LANGUAGE-RELATED HINTS	291
<i>Formatting Multiple Choice and True/False Questions</i>	291
<i>Fixed Width Blanks in Cloze Exercises</i>	291
<i>Default Font Attributes for Answers in Cloze Exercises</i>	291
ARABIC AND HEBREW (RIGHT-TO-LEFT) SCRIPTS REFERENCE	292
<i>Right-to-Left and Left-to-Right Languages Basic Information</i>	292
CHINESE	293
<i>The Pinyin Phonetic Input Method</i>	293
<i>Conversion Preference</i>	293
<i>To Convert or Not to Convert?</i>	293
<i>Watch Your Tones!</i>	293
<i>What is Your Pinyin Status?</i>	294
<i>Functions of the Space Bar</i>	294
<i>Selecting Options from the Chinese Selection Dialog Box</i>	294
<i>Chinese Function Keys</i>	295
<i>Chinese Status Bar</i>	296
<i>Special Pinyin Forms</i>	297
<i>National Phonetic Alphabet ("Zhuyin Fuhao" or Bopomofo)</i>	298
<i>Zhuyin Fuhao-Pinyin Correspondence Table</i>	298
<i>The Cangjie Input Method</i>	298
KANGXI RADICAL CHART	301
JAPANESE	304
<i>Keyboards</i>	304
<i>Peculiarities of Typing in Japanese</i>	305
<i>Kana Kode Key</i>	308
KOREAN	314
<i>How to Type Hangul</i>	314
<i>Hangul-Hanja Conversion</i>	316
<i>How to Type Korean Transliteration</i>	317
CHAPTER 10: CONFIGURING AND CUSTOMIZING WINCALIS AND WINCALIS AUTHOR	319
THE MAIN WINCALIS INITIALIZATION FILE WINCALIS.INI	319
<i>General Tips</i>	319
<i>Changing the Default Font Size for Use in WinCALIS Author</i>	320
<i>Setting the Virtual Keyboard Type</i>	321
<i>Setting the WinCALIS Startup Mode (*.CAL or *.WCL)</i>	321
<i>Setting WinCALIS Window Defaults</i>	321
<i>Changing the WinCALIS Author Desktop</i>	322
<i>Setting the Top and Initial Languages in your Language Menus</i>	323
<i>Rearranging the Language Menus</i>	323
<i>Setting Default Directories</i>	325
<i>Defining Language Keyboard Files</i>	330
DELETING NONESSENTIAL FILES TO SAVE ON DISK SPACE	341
INSTALLING MCI DRIVERS FOR MULTIMEDIA	342
<i>Installation in Control Panel</i>	342

<i>Making MCI Drivers Work in WinCALIS</i>	342
<i>Other MCI Information</i>	343
INSTALLING WINCALIS ON A NETWORK	343
<i>WinCALIS Home Directory</i>	343
<i>Writing *.WCL Files</i>	343
<i>Double dots “..”</i>	343
<i>Non-ASCII Characters in DOS filenames</i>	344
<i>Tips on Installing WinCALIS on a Shared Network Drive from a Local Workstation</i>	344
HOW TO UNINSTALL WINCALIS AND WINCALIS AUTHOR	345
<i>Uninstalling Using the Windows File Manager:</i>	345
CHAPTER 11: WINCALIS CODE AND COMMAND CONVERSIONS	347
<i>Eight-bit Code Conversions</i>	347
<i>Sixteen-bit Code Conversions</i>	350
<i>Twenty-four-bit Code Conversions</i>	355
<i>UTF-7 and UTF-8 Conversions</i>	356
<i>Conversion of Commands in Older WinCALIS Scripts</i>	357
APPENDIX A: CALISSCRIPT LANGUAGE QUICK REFERENCE	359
STATEMENTS	359
WILD CARDS	359
OTHER RESPONSE LINE SYMBOLS	359
SPECIAL RESPONSES	360
COMMANDS	361
GLOBAL VARIABLES	364
OPTIONS	365
APPENDIX B: SELECTED KEYBOARD LAYOUTS	367
APPENDIX C: GLOSSARY	377
INDEX OF FIGURES	395
INDEX	397

About The WinCALIS Author User's Manual

To accompany the recent flourish of improvements in WinCALIS technology, we are proud to offer a new and expanded version of the WinCALIS documentation. In the following paragraphs, the general layout and methodology employed will be explained.

Let us start with some good news: WinCALIS is still easy to use! With extensive on-line help, answers to your questions are never more than a mouse-click away. You will find step-by-step instruction *and* reference information all in one place, thus providing the practice that beginners need and the handiness that advanced users prefer.

Who You Are and Where to Go

In the beginning sections, step-by-step examples will get you started with hands-on use. As your proficiency increases and the topics become more complex, experimentation will be encouraged, specific to your personal application. Throughout the documentation, examples, figures, and flow charts will guide you through the text.

Beginning Users

If you are just beginning with WinCALIS, we suggest that you start, naturally, at the beginning. New users are usually anxious to get results quickly, so we've designed things to have you using and implementing the software from the start. But we caution you to advance at a rate which will insure a grasp of the fundamentals. In short, walk before you run.

Intermediate and Advanced Users

For those of you with some WinCALIS experience, we invite you to review or browse through the first few sections, which cover some fundamentals, and then to concentrate your study beginning around the section Customizing and Communicating. Complex features will be explained in great detail, as well as audio, graphics and mouse implementation. Of great significance is the section The CALIScript Language, in which the CALIS authoring language is explained.

Note that any example CALIS lesson (in *.CAL file format) may be copied from the on-line help, pasted and saved in the WinCALIS Author Scratchpad, and run in WinCALIS.

Section Summaries

Introduction: What is CALIS?

The introduction explains the basic concept behind WinCALIS, from its strengths to its structure. You'll also find important information on how to contact us at the Humanities Computing Laboratory.

Getting Started With WinCALIS

This section will move you right along from the introductory demonstration lesson through the WinCALIS menu arrangement to working with the screen elements and producing accents and diacritics.

Authoring Basics

This section will introduce you to the creative process of authoring, in which you design computer lessons for your students. You will create your first lesson with our help, and come to understand the WinCALIS Author windows arrangement and many of the menu items.

Lesson Development

This section discusses the “nuts and bolts” of question and feedback development, the area of lesson design which is most instructive and helpful for students. Methodology and implementation are explained for the various WinCALIS Author question types, as well as the options which govern how WinCALIS and WinCALIS Author work mechanically with you and the student.

Customizing and Communicating

This section brings some of the more fun elements of lesson design into the foray, treating issues of aesthetics and feedback flexibility. Along with explanation of extra-CALIS file use, this Section concentrates on how to communicate attractively with your output and flexibly with student input.

The CALIScript Language

This section takes you deeper into the CALIS system as you get your feet wet with CALIScript, the CALIS authoring language which constitutes the heart of CALIS. You will learn how to alter and improve your lessons by using language in conjunction with WinCALIS Author.

Multimedia in WinCALIS

This section tours the Multimedia Editor as a magnificent multimedia tool for audio and video authoring. Then it shows you how to insert the audio and video in your lessons. You will learn about the necessary hardware setup and WinCALIS commands for using an audio card, digital video, laserdisc player, or VCR to bring live action to your lessons.

Special Authoring and Editing Functions

This section is an ad-hoc catchall. Included is information about on-line user dictionary use and creation, student record-keeping, and printing.

New Features in WinCALIS

New features and enhancements are constantly being added to WinCALIS. To help you discover what's new, and to make the transition for those items that are too new to have been fully integrated with the rest of the manual, we offer this section.

Specific Language Reference Manuals

This section is dedicated to the special commands, options, and keyboard shortcuts for specific languages, especially for the “CJK” languages--Chinese, Japanese, and Korean--and the right-to-left script languages--Hebrew, Arabic, Persian (Farsi), Urdu, Yiddish, etc. Contact us at our address listed in this Introduction and in the section of On-line Help “What is CALIS” for more information on purchasing the CJK (Chinese-Japanese-Korean) module or the Right-to-Left Scripts module.

Configuring and Customizing WinCALIS Author

This section gives pointers on how to customize your copy of WinCALIS Author so that it best serves your individual needs.

WinCALIS Code and Command Conversions

This section deals with information necessary to convert files to and from WinCALIS 2.x Unicode and other coding standards, including older versions of text CALIS and WinCALIS 1.x files. It also suggests some techniques for fine-tuning older scripts, including adapting commands to WinCALIS 2.x.

CALIScript Language Quick Reference

This section acts as a quick reference guide to WinCALIS statements, commands, options, and special responses.

Introduction: What is CALIS?

CALIS is the “Computer-Assisted Language Instruction System.” It combines the most innovative concepts of Computer-Assisted Learning with the soundest pedagogical principles to equip language teachers with a stimulating educational tool. With CALIS, you can prepare language lessons which supplement classroom learning, i.e. create exercises to be administered, corrected, and scored individually for each student. With the successful integration of the popular Microsoft Windows environment, our text-based version is now available, new and improved, as WinCALIS. Successful language education now comes with the click of a mouse.

Using your own creativity and pedagogical prowess, WinCALIS lets you design and implement a modular lesson program which can teach and test simultaneously. Students complete the lessons at their own speed and receive instant feedback on their work. The classroom, language lab, and computer lab thus come together to give the student and yourself the best mix of computer-assisted learning.

The Strengths of WinCALIS

Easy to Use

Microsoft Windows and a friendly user environment allow the language instructor to create lessons quickly and easily. Likewise, students have been impressed by the simplicity of using WinCALIS.

Multilingual

WinCALIS can communicate in just about any language. Presently supporting hundreds of languages, written in more than twenty orthographic scripts (see below for sample), the open-ended design of WinCALIS technology means the sky's the limit in terms of linguistic capability. We are working in conjunction with our users to develop the necessary fonts, character sets, and keyboards to support even more languages.

Some of the Languages Supported by WinCALIS

Amharic (Ethiopic)
Arabic
Armenian
Belarusian
Bulgarian
Chinese
Croatian
Czech
Danish
Dutch
English
Finnish
French
Gaelic
Georgian
German
Greek
Hausa

Hebrew
Hungarian
Icelandic
Indonesian
Italian
Japanese
Korean
Lappish
Latin
Latvian
Lithuanian
Macedonian
Malay
Mongolian (Cyrillic)
Norwegian
Old English
Persian (Farsi)
Phonetic (IPA)

Polish
Portuguese
Romanian
Russian
Serbian
Slovak
Spanish
Swedish
Tagalog
Thai
Turkish
Ukrainian
Urdu
Vietnamese
Yiddish
Yoruba

Flexible

The language instructor has complete control over what the student sees and does, but WinCALIS also provides the option to use any of the built-in lesson formats. A wide variety of textual display and scoring modes, along with multimedia visual and audio capability, give the author plenty of options. Specific keyboard mappings are presented as defaults but may be customized.

HELP-ful

On-line help means you are never more than a mouse-click away from the answer to your question. Complete with commands and menus, the Windows help system is fast and informative.

Stimulating

The short modular lesson format gives instant feedback, and incorporation of multimedia makes lessons more interesting and therefore more educational for your students.

Multimedia

On-line graphics are easily incorporated into every lesson to help *demonstrate* your lesson objectives. In addition, the Multimedia Editor allows you to utilize live action (audio and visual) in lessons.

Structure

The WinCALIS author workstation package consists of five main modules: the WinCALIS program itself, the WinCALIS Author, the Multimedia Editor, the Multilingual Printer and the TrueType Printer. All five components come under the rubric of "WinCALIS Pangaea."

The WinCALIS component is that part of the WinCALIS package which actually runs the lessons. The lessons themselves may be created quickly and efficiently by WinCALIS Author. After a lesson is created in Author, it is ready to be run by WinCALIS. The Multimedia Editor allows you to insert audiovisual clips from digital video files, laserdiscs, and VCRs (provided you have the necessary device drivers or hardware--see the section Multimedia in WinCALIS). Finally, the WinCALIS printers allow you to print Text files complete with special CALIS coding information. Put together, these five components form an easy to use, multilingual, multimedia language instruction tool.

Note to Users of Text-Based CALIS

Fortunately, most, if not all of your text-based CALIS lessons ("scripts") are fully executable under WinCALIS. The CALIS programming language and the ASCII font mappings are essentially the same, so WinCALIS accepts input and produces output in the same basic manner.

The real power of the WinCALIS authoring program, WinCALIS Author, is the ease and speed with which you can make scripts *without directly using the CALIScript Language*. You provide WinCALIS Author with the text, directions, questions, screen layout, etc., and it writes the script for you. Of course, very advanced users may still want to tinker with a script using a word processor to make changes or additions.

The overall file structure for scripts has also been modified, but is fully interactive with the text-based CALIS file structure. In text-based CALIS, as you recall, scripts are saved with the *.CAL extension and consist of ASCII and IBM-extended ASCII text, written in CALIScript. WinCALIS can also use the same format of ASCII/ANSI text file thus allowing for script modification with a text editor.

Hardware and Software Requirements

- A personal computer with an Intel 80486 or higher processor, a minimum of 16MB of RAM, and standard VGA or higher graphics.

- A hard disk with 25MB of free disk space (100MB for a full installation) and a CD-ROM drive or network connection for installation.
- Windows 3.1 (WinCALIS ver.2.3) or Windows 9.x/NT/2000/XP (WinCALIS v.2.5).

See the section Multimedia in WinCALIS for Multimedia Editor requirements.

Copyright Information

© Copyright 2001 by the Humanities Computing Laboratory
Durham, North Carolina, USA
All rights reserved.

No part of this publication may be reproduced, transmitted, or transcribed, stored into a retrieval system or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written permission of the Humanities Computing Laboratory.

The Humanities Computing Laboratory reserves the right to revise this publication and to make changes from time to time in the content of this publication without obligation of the Humanities Computing Laboratory to notify any person of such revision.

Microsoft is a registered trademark of Microsoft Corporation; Windows operating systems are trademarks of Microsoft Corporation.

IBM is a registered trademark of International Business Machines Corporation.
Sound Blaster is a trademark of Creative Labs, Inc.

Pioneer and Sony are registered trademarks of their respective corporations.

Autodesk Animator is a trademark of Autodesk, Inc. Gold Disk is a registered trademark of Gold Disk, Inc.

WinCALIS Registration

Be sure to fill out and mail the registration form so we can alert you to all new WinCALIS developments and opportunities.

Send your completed WinCALIS Registration Form to one of the following addresses:

In the USA and non-Argentina:

Humanities Computing Laboratory
301 W. Main St., Ste. 400-I
Durham, NC 27701 USA
Phone: (919) 667-9556, (919) 656-5915
Fax: (919) 667-9556
E-mail: info@humancomp.org
Web: <http://www.humancomp.org>

In Argentina:

Centro Blas Pascal

Investigación Y Desarrollo En Informática Educativa
Bacacay 705 - piso 11 "B"
Ciudad de Buenos Aires
Buenos Aires, Argentina
Voice: +54 1 4432-8353 / 8358
Fax: +54 1 4432-8353 / 8358 (15) 4181-9904
bpascal@ciudad.com.ar

You must be a registered user to access Humanities Computing Laboratory's or its agents' Support Services.
Please make sure to mail or fax your registration form today.

WinCALIS Support Services

You may contact the Humanities Computing Laboratory or its agents for Technical Assistance regarding the functionality of the program, its subset of commands, and the way the software works with the hardware, all as stated in the documentation.

No matter how you choose to contact us, please be prepared to provide the following information:

- Your Serial Number
- Product version number listed found by choosing About from the Help menu in WinCALIS or WinCALIS Author.
- The type of computer hardware you are using.
Optionally, and according to the type of question you may have, also provide us with the following:
 - Software version of Microsoft Windows.
 - Contents of the WINCALIS.INI file.
 - Exact wording of any messages on the screen.
 - What you were doing when the problem occurred.
 - How you tried to solve the problem.

Electronic Mail

Ask questions and receive help from Technical Support via e-mail, which is by far the most flexible way of getting support. Just send the electronic mail to the accounts listed below. This option is given the highest priority.

(USA) info@humancomp.org
(Argentina) bpascal@ciudad.com.ar

World Wide Web

In order to facilitate the scholarly exchange of WinCALIS-related data, we maintain a World Wide Web site. Please point your WWW browser to <http://www.humancomp.org>.

Fax

Feel free to send us your product orders, questions or comments. Our fax numbers are:

In the USA: 1-919-667-9556

In Argentina: +54 1 4432-8353 / 8358 (15) 4181-9904

Telephone

If you would like to talk to us in person, or if you have questions you would like to ask us, use these telephone numbers. You can reach Technical Support Services by phone between 9:00 A.M. and 6:00 P.M. ET, Monday through Friday. When you call, you should be at your computer and have your documentation at hand. Be prepared to provide the information as requested on the previous page.

In the USA: 1-919-667-9556 or 1-919-656-5915

In Argentina: +54 1 4432-8353 / 8358

Address

Please feel free to write us at the following addresses:

In the USA and non-Argentina:

Humanities Computing Laboratory

301 W. Main St., Ste. 400-I

Durham, NC 27701 USA

In Argentina:

Centro Blas Pascal

Investigación Y Desarrollo En Informática Educativa

Bacacay 705 - piso 11 "B"

Ciudad de Buenos Aires

Buenos Aires, Argentina

<p>You must be a registered user to access Humanities Computing Laboratory's or any of its agents' Support Services. Support Services are subject to prices, terms, and conditions in place at the time the service is used.</p>

Chapter 1:

Getting Started with WinCALIS

System Requirements

Before you begin the installation process, make sure that you have all the hardware and software you need to run WinCALIS. At minimum, you will need the following:

- A personal computer with an Intel 80286, 80386, or higher processor and a minimum of 2M memory running in standard or enhanced mode.
- A graphics adapter and monitor supporting one of the following systems: EGA, VGA, 8514/a, or Hercules Graphics card.
- A hard disk with 4M of free disk space and one of the following: 3 1/2" (1.44M) disk drive or 5 1/4" (1.2M) disk drive.
- Windows 3.1 or higher running on your machine.

Note: Because some WinCALIS features are not easily accessible from the keyboard, the use of a mouse is recommended.

Installation (WinCALIS)

The files on the WinCALIS program diskettes are in compressed format. The Setup Program decompresses the files as it copies them to your hard drive. WinCALIS cannot be run from the diskettes included in your package. You must first install the software using the WinCALIS Setup program.

With Windows running on your machine, insert the first setup disk into the 3.5" floppy disk drive.

- In Windows 3.x: From the Program Manager (not File Manager), click on the File menu item, then select Run.
- In Windows 95, click the Start button, then select Run.

You will then see a Run dialog box. In the Command Line, type **a:\setup** (substituting 'a' for the letter of your 3.5" drive). Press Enter or click the OK button. This will start the installation program.

Note: You may exit at any time during the setup procedure.

The first thing you will be asked to do is enter your name and organization or affiliation. When you press Enter, you will proceed to the next screen which will verify that the organization name you just entered is correct. Next, setup will inform you of the file(s) that it will install automatically.

Next, setup will assign a default drive and directory where the programs will be installed, unless you specify an alternate drive and/or directory, by clicking on the Set Location button. You will also be provided with information on space required and space available on the hard drive.

Note: If you click on install at this point only WinCALIS will be installed. To install additional files click on the Optional modules button. Optional modules include multimedia, graphics, additional language scripts, etc.

Choose Optional Modules

If you clicked on the Optional modules button, a dialog box will appear listing various module options. You can choose to install only that part of WinCALIS you will be working with by clicking on the desired modules with the mouse (hold down the Ctrl button to add to your highlighted selections), or click the All button. When you have made your selections, press the Continue button. If the All button is chosen, Setup will install all modules available in WinCALIS. In order to install the optional modules, you must have the corresponding setup disks.

Once the modules are chosen, click on the Install button. WinCALIS will verify that you have sufficient memory on the hard drive before beginning installation.

An overview of WinCALIS fundamentals is available as a part of the installation package.

WinCALIS Structure

If you are unfamiliar with WinCALIS, you are encouraged to take the time to run through the sample lessons included as part of the standard Author Workstation. Feel free to run the Author program and open and browse through the corresponding *.WCA script database files, which are included for some of the sample lessons.

You should run through the demonstration lesson or one of the other WinCALIS sample lessons, showing how the program operates from the perspective of the student. As outlined in the section What is CALIS?, a WinCALIS author workstation consists of five main components: one for running lessons, WinCALIS; one for making lessons, Author; one for incorporating multimedia, the Multimedia Editor, and two printer programs, WinCALIS Multilingual Printer and WinCALIS TrueType Printer. There are also other supplementary files and components.

Starting WinCALIS

We will now examine how to start WinCALIS, move around in the WinCALIS environment, and work a WinCALIS lesson. If you are unfamiliar with Microsoft Windows, please go through the demonstration program provided for you on Microsoft Windows or check your Microsoft Windows User's Guide.

After you have Windows up and running on your machine and have installed WinCALIS, select the WinCALIS program icon to start it. You will be presented with the opening screen, welcoming you to WinCALIS. At this point you have some decisions to make about the directory location and format of the lesson you wish to run, and how you want the screen to look. As explained below, you use the Options choice on the main menu bar. To help you understand all of your options, the on-line help system is always available. After you have things the way you want them, you can select and run the desired lesson. Any changes you make in the menu options will be in effect until you change them again, so you do not have to reset them every time you run WinCALIS.

All of these features and decisions are controlled by menu items on the main menu as explained in the section WinCALIS Opening Screen Layout and Menus

WinCALIS Opening Screen Menus and Layout

Menu Bar

Just beneath the WinCALIS title bar with the universal Windows icons, you will find the menu bar with Options, and Help.

Options Menu (Author Version of WinCALIS Opening Screen)

No Conversion (*.WCL)

As mentioned in the section Note to Users of Text-based CALIS, WinCALIS can generally update CALIS scripts without too much difficulty. [See the section WinCALIS Conversions for examples of problem situations which may require editing of the older scripts.]

Only those lessons in the current directory in the WinCALIS binary file format (with the extension .wcl) will be displayed in the Lesson Window. Lessons created by WinCALIS Author are automatically available in this format.

Text Conversion (*.CAL)

Only those lessons in the current directory in Unicode text-based format (extension .cal) will be displayed. These lessons may be edited using WinCALIS Author Scratchpad (see the section "CALIScript"), and are converted to the WinCALIS Unicode binary-file format (extension .wcl) when run.

Change Directory...

Allows you to access a lesson by specifying the entire pathname, including the drive letter, the directories and sub-directories, and ending with the name of the WinCALIS lesson. (The name of the WinCALIS lesson will have a suffix, either *.wcl or *.cal as explained in the Conversions section below.

Window Defaults...

Specifies the aesthetic qualities of the screen in each of the various windows that WinCALIS employs. For each window that WinCALIS utilizes, you may specify the text color, text attributes, or style (e.g. **bold**, *italic*, etc.), point size, and background color. Each WinCALIS window (Directions, Text, etc.) may have different characteristics, as selected by the pull-down menus in Window defaults.

Note: The option to switch between "No conversion" and "Text conversion" and the option to set "Window defaults" are found only on Author Workstation versions of WinCALIS.

Help Menu (WinCALIS Opening Screen)

As noted in the introduction, WinCALIS comes with an extensive on-line help system. You will soon learn how to find your answer quickly, using on-line help in conjunction with the printed manual. Clicking on the Help menu, pressing F1 or pressing Alt+H will get you the help you need instantly.

Index

The index provides the user with several categories of help topics ranging from "Choosing a lesson" to "Working efficiently." These topics function as mini-lessons about WinCALIS. To get help on more specific topics, such as Commands, use the Windows Help Index, along with the Search and Browse features.

Selection Window

This topic explains the WinCALIS opening screen and lesson selection window.

Text-CALIS/WinCALIS 1.x Conversion

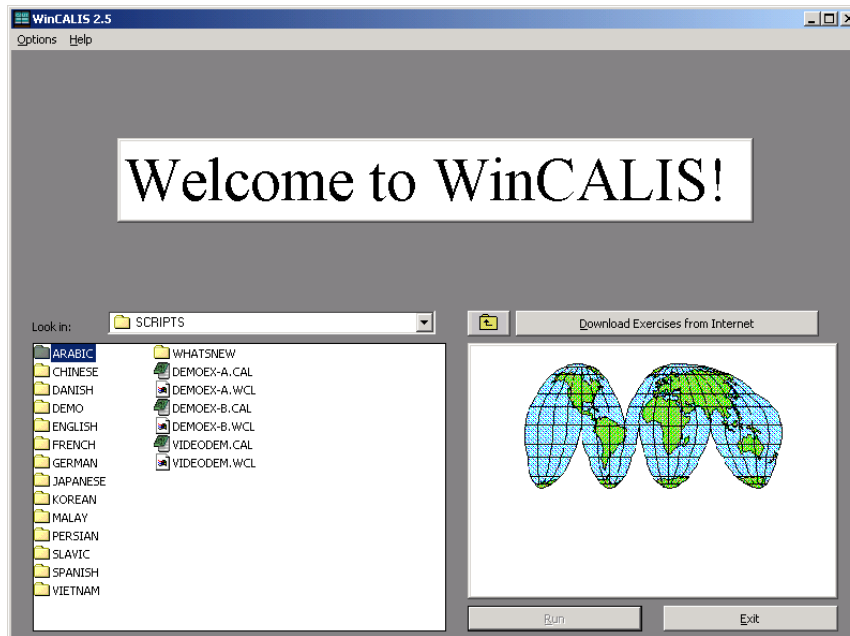
This special help topic gives information about the conversion of scripts made for older versions of Text CALIS (DOS IBM-extended ASCII), WinCALIS 1.0 (mixed DOS/OEM and ANSI), and WinCALIS 1.1 (ANSI).

This is essential reading for anyone updating old files.

About WinCALIS...

About WinCALIS shows the current version and date of your copy of WinCALIS. It also displays the serial number of the package and the name it is registered to.

WinCALIS Opening Screen Layout



The screen is divided into three main areas beneath the menu bar, each communicating information about lessons and the directories in which they reside. Each area is described below.

Lesson Window

In the lower-left part of the screen, beneath the “Look in: [directory path]” message, you will find an alphabetized list of all the WinCALIS lessons and folders containing more lessons in the selected folder. You may use the directional arrows or click the mouse to highlight the lesson. Press ENTER or double-click to start the lesson. You may also click on the Run button below the Lesson Window. WinCALIS is installed during Author Workstation Setup so that it starts automatically with the Lesson Window showing all applicable files (*.CAL, *.WCL, and *.ZIP). This view can be changed in the Options menu.

Lesson Description (Comment) Window

For each lesson in the Lesson Window there may be an associated message or statement displayed in the Lesson Description Window. It usually summarizes the content or objective of a lesson, helping students to prepare mentally before starting the lesson. Other information, such as a reminder to the student about tutorials available in the lesson, can also be displayed. If the author does not specify a default lesson description, the message “Welcome to WinCALIS” will be displayed.

Note: You may change the default setting so that WinCALIS starts up on your Author Workstation in several different display modes. Use the Windows Notepad or other plain-text editor to edit the “ListBoxDisplay=” line in the [WinCALIS] section of the WinCALIS initialization file WINCALIS.INI, in the main WinCALIS program directory (usually the parent directory of the default scripts directory). The values are as follows:

ListBoxDisplay=0	(shows *.WCL only --No Conversion)
ListBoxDisplay=1	(shows *.CAL only -- Text Conversion)
ListBoxDisplay=2	(shows *.ZIP only -- Archive Conversion)
ListBoxDisplay=3	(shows all applicable files *.WCL, *.CAL, and *.ZIP).

Command Buttons

You may click on Run to begin the selected lesson or click on Exit to exit WinCALIS and return to the Windows Program manager. You may click on the Up Arrow button to go up one note to the parent folder in the directory tree. You may click on the “Download Exercises from Internet” button to go to a World Wide Web or FTP site from which lessons may be downloaded, normally packaged in compressed .ZIP archive files. Once lesson is selected and downloaded, the name will appear in the Directory Window on the lower left. It can then be run by clicking on the lesson name there.

Directory Description Window

In the lower right-hand corner, you will find the Directory Description Window, where the author may display a message concerning the current directory highlighted in dropdown box above the Lesson Window. Authors can convey preliminary instructions, objectives, or other information about the group of lessons in the highlighted directory. This is accomplished by placing a Unicode text file in the directory called “WINCALIS.DIR,” containing the information to be displayed in the Directory Description Window. This file may contain formatting commands, graphics, and most other CALIS commands, as long as they do not reference another window.

Lessons should normally be grouped together by related subject matter. For example, the directory, “Pronouns,” might contain five or six different lessons based on the various forms of the language's pronouns. Additional information such as due dates for the lessons could also be displayed in the Directory Description Window.

Note: If there is no “WINCALIS.DIR” file in the highlighted directory, then the WinCALIS Pangaea world map image will be displayed.

Working a WinCALIS Lesson

If you have not completed the demonstration lesson or another WinCALIS sample lesson, it is suggested that you do so. There is no better way to learn how to use WinCALIS. If you have completed the lesson, the following will be primarily for review and reference. We will begin with the menu items.

WinCALIS Menus

Lesson Menu (WinCALIS)

[L]esson] Quit Lesson

Terminates the current lesson and sends the user back to the WinCALIS opening screen in order to select a new lesson.

[L]esson] Save Window Contents As...

Allows you to save the contents of the selected window to a file on disk.

[Lesson] Exit WinCALIS

Terminates the WinCALIS program altogether.

Options Menu (WinCALIS)**[Options] Add Accent F2**

Selecting the menu item “Add Accent,” or pressing the shortcut key F2, will highlight the letter to the left of the insertion point (with blinking caret), and the next letter pressed will overstrike the highlighted letter as a diacritical modifying accent (grave, acute, circumflex, tilde, diaeresis, etc.). See the section Accents and Diacritics.

[Options] Switch Languages F3

Selecting the menu item “Switch Languages,” or pressing the shortcut key F3, will switch between the currently selected language and the top menu language (by default English/W. Europe, the ANSI/Latin1 keyboard).

Note: The “top” language is determined by the default language .INI file defined in the “DefaultIni=” line in the [WinCALIS] section of the main WinCALIS initialization file “WinCALIS.ini,” in the WinCALIS program directory. The default language .INI file is set by default for the United States and Western Europe to “latin1.ini.” If the default .INI file is changed to some other language .INI file, the “top” language will also be changed accordingly. The initial language for use on startup in WinCALIS and WinCALIS Author (also by default set to English/W.Europe) is also customizable by the courseware author or WinCALIS system administrator. The initial language is determined by the setting of the InitialLanguage= line in the [Initial] section of the “WinCALIS.ini” file. The out-of-the-box value for this line is InitialLanguage=English/W.Europe.

[Options] Give Up F6

By choosing the Give Up option or pressing F6, a student may (in some circumstances) give up when he or she does not know the answer. The answer designated as correct/continue will then be displayed. By default, a user may not give up on a particular answer until after attempting the question at least once. Students are generally discouraged from giving up too soon, so that they are challenged to rethink their answers. Also WinCALIS gives users the option of passing a question and returning to it later. (See Pass Back, Pass Forward below.)

[Options] Directions F9

By choosing the Directions option or pressing F9, the student may call the Popup (Directions) Window to the screen.

[Options] Play Audio F11

When prompted to do so by the directions, the user may play an audio clip provided by the author of the lesson.

[Options] Play Video F12

When prompted to do so by the directions, the user may play a video clip provided by the author of the lesson.

[Options] Unicode Browser...

Opens the Unicode Character Selection dialog box, which allows the student to browse through the Unicode character tables. The student may select a character by its Unicode character code, either by scrolling through Unicode character blocks and selecting the desired character, or by typing the hexadecimal Unicode character code in the box in the lower right corner.

See: Unicode Character Selection Dialog Box.

[Options] Media Controls (MCI Control Bar)

At the author's discretion, a control bar is displayed for the student to control the playback through the Windows Media Control Interface (MCI) of the current, or most recently played, audio or video clip. The student can suppress the display of this control bar by deselecting it here. If there is no audio or video available, the menu item is dimmed.

[Options] Pass Back Gray - and Pass Forward Gray +

By choosing these options or pressing the gray “+” or “-” keys, you may skip back and forth through questions. Instead of giving up, if you do not know the answer to a question, you can go on and answer the succeeding questions, which may provide a clue to answering the preceding one.

Note: As a convenience for laptop users, WinCALIS also supports the additional Pass Forward key combo **Ctrl+PageDown** and the additional Pass Back key combo **Ctrl+PageUp**.

Tutorials Menu (WinCALIS)

The author may include supplementary instructions with the lesson to help the user answer the questions. This information is accessed by the user via the Tutorials menu.

By clicking on the Tutorials menu or pressing Alt+T, a list of all tutorials will be displayed, along with their access keys. Double-clicking on the specific tutorial or pressing Shift and the appropriate function key will open the window. Tutorials give the author of the lesson a way of giving “hands-on” assistance to the user.

Language Menu (WinCALIS)

It will frequently be necessary for the user to select a language keyboard other than the default keyboard. He can choose the language keyboard he prefers from the Language menu.

The author should remind the student to do this in the directions for the lesson, unless the student is already familiar with the procedure from previous lessons. For example, if the author requires an answer in Arabic, the author can tell the student in the directions to go to the Language menu and select his preferred language keyboard for Arabic. Or the author can preset the language keyboard for the student with the “Use [English/W.Europe] Keyboard” option in the Script Generation Options dialog box.

When the language pull-down menu is opened, all supported language character sets will be displayed. Language keyboards are grouped into sub-menus either by geographic region (e.g., Europe other, Japanese) or by script family (e.g., Arabic, Cyrillic, Indic). By double-clicking on a language or highlighting it and pressing Enter, you determine the key-to-character correspondence of the keyboard. A virtual keyboard can be displayed on screen by clicking on Keyboard help in the Help menu. For an explanation of conventions used in the virtual keyboard, see Accents and Diacritics. Note that by pressing the shortcut key F3 or selecting the “Switch Languages” option under the Options menu, you can toggle between the currently selected keyboard and the top keyboard (usually “English/W. Europe”).

Western European languages such as Danish, English, French, German, Italian, Portuguese, Spanish, and Swedish are all available with the ANSI (Latin1) Roman character set and default “English/W.Europe” keyboard (no keyboard change required). All accents are typed with three keystrokes (base letter + F2 + accent).



Figure 2: Keyboard Help

Remember that the open-ended, multilingual design of WinCALIS means that any language can be supported if the font has been developed. With the input of our users, we have been gradually adding languages. Please contact us for possible collaboration on new languages. See the section What is CALIS? for our address and phone numbers.

See Also: L: [Language] Menu (WinCALIS Author)

Specific Language Reference Manuals

Switch Languages.

Right-to-Left and Left-to-Right Languages

Tools Menu (WinCALIS)

[Tools] Language Options

Editing options which are specific to particular languages are grouped under this menu item. Each language has its own cascading submenu.

[Tools, Language Options] Chinese Options

Reselect ("Ops")	F9
Character Search	F7
All-tone Search	F8
Default Selection/Repeat	F10
Set Cache/No-Cache Search Mode	Alt+F3
Set Convert/No-Convert Mode	Alt+F5
Set Word/Character Search Mode	Alt+F7
Set Tone/All-Tone Search Mode	Alt+F8
Set Simple/Complex Style	Alt+F9
Set Default/No-Default Search Mode	Alt+F10
Set Word/Character Search Mode	Alt+F7

These menu options alter the Chinese pinyin-Hanzi conversion process. For an explanation of the various Chinese conversion settings and search options, see the topic Chinese Function Keys or the general Chinese Language Reference (requires separate CJKV module).

More Options... Alt+F11

If the current language keyboard is Chinese Pinyin-to-Hanzi, selecting the menu item Tools, Language Options, Chinese Options, More Options will display the Chinese Dictionary Search Options dialog box, with the current settings for Chinese input. These settings can be changed either by clicking with the mouse or by pressing the corresponding Alt + function key combination.

Set Disk/No-Disk Search Mode Alt+F12

As with the other menu options above this option alters the Chinese pinyin-Hanzi conversion process. See the topic Chinese Function Keys or the general Chinese Language Reference (requires separate CJKV module).

Chinese Status Bar

When you choose Chinese Pinyin-to Hanzi as the active language keyboard, the Chinese Status Bar will pop up by default, displaying the current settings for Chinese input.

The Status Bar may be turned off by deselecting it here. If Chinese Pinyin-to-Hanzi is not the currently selected language, the menu item is dimmed.

See: Chinese Status Bar (requires separate CJKV module).

[Tools, Language Options] Japanese Options

View Japanese Dictionary...

Allows you to view and edit the Japanese internal Kana-Kanji conversion dictionary. (Currently disabled.) See: Japanese (requires separate CJKV module).

[Tools, Language Options] Korean Options

Hanja-to-Hangul F7

To convert the syllable or word to the left of the insertion point (the blinking caret) written in Hangul into its Hanja equivalent, press the F8 function key, or click on the menu item Tools, Language Options, Korean Options, Hangul-to-Hanja.

Hangul-to-Hanja F8

To convert the syllable or word to the left of the insertion point (the blinking caret) written in Hanja into its Hangul equivalent, press the F7 function key, or click on the menu item Tools, Language Options, Korean Options, Hanja-to-Hangul.

See: Korean (requires separate CJKV module).

[Tools] Look Up Word... F5

If an on-line dictionary has been installed on the WinCALIS workstation, selecting this menu option will open a dialog box, allowing the user to look up the meaning or pronunciation of a selected word or phrase. Depending on how the dictionary was installed, it is either accessible only from specific language keyboards (e.g. for a Korean-English dictionary, the current keyboard must be one of the Korean keyboards), or accessible from any language keyboard which does not have a specific dictionary linked to it (e.g. for a Korean-English dictionary, the current keyboard could be either the default English/W. Europe keyboard or any of the Korean keyboards).

In order to select a word to look up, follow one of the following three methods.

- Holding down the right mouse button, drag the mouse across the word you wish to mark for looking up. The marked word will appear highlighted in a pop-up box in the lower part of the screen. Press the space bar or click the mouse in the box to close it, then press F5 or click on the menu item Tools, Look Up Word... to look up the marked word.
- Copy the entire contents of a WinCALIS window directly to the "Dictionary Mode Window," by selecting the menu item Tools, Dictionary Mode Window. First you will be asked to select the window to be copied. Then the WinCALIS Composition Editor will automatically be launched with the clipboard text pasted into the Dictionary Mode Window, ready for looking up. Drag the mouse across the word or phrase to be looked up (or double click on it if it is a single word) to highlight it.

Then press F5 or click on the menu item Tools, Look Up Word... in the Composition Editor to look up the word. Repeat as often as desired. To return to WinCALIS, either exit the Composition Editor or leave it running for further use and switch to WinCALIS by using the Windows shortcut key Alt+Tab.

- Copy the entire contents of a WinCALIS window to the Windows clipboard by selecting the menu item Tools, Copy Window to Clipboard. Then use Alt+Tab to switch to the WinCALIS Composition Editor (a special version of UniEdit). If this is not yet running, launch it by selecting the Composition Editor menu item from the Tools menu. Paste the clipboard contents into any document window. Then proceed as in the previous method to drag the mouse across the word or phrase to be looked up (or double click on it if it is a single word) to highlight it. Then press F5 or click on the menu item Tools, Look Up Word... in the Composition Editor to look up the word. Repeat as often as desired. To return to WinCALIS, either exit the Composition Editor or leave it running for further use and switch to WinCALIS by using the Windows shortcut key Alt+Tab.

You can press the Escape key at any time during a search to cancel further searching and display the matches found thus far.

See Also: User-Supplied On-line Dictionaries

[Tools] Dictionary Mode Window...

Selecting this item launches the WinCALIS Composition Editor (a special version of UniEdit) and copies and pastes the text contents of a selected WinCALIS window to a “Dictionary Mode Window,” from where words can be easily highlighted and looked up in an on-line dictionary. A Select Window dialog box allows you to choose which window’s contents to copy and paste.

If the WinCALIS Composition Editor has already been launched and is currently running in the background, use the Tools, Copy Window to Clipboard” menu item to copy a window the clipboard, then switch to the Composition Editor with the Windows shortcut key Alt+Tab, and paste the window contents there. Then proceed to highlight and look up desired words there.

[Tools] Copy Window to Clipboard...

Selecting this item copies the text contents of a selected WinCALIS window to the Windows clipboard. A Select Window dialog box allows you to choose which window’s contents to copy.

[Tools] Composition Editor

Selecting this item launches the WinCALIS Composition Editor (a special version of UniEdit).

[Tools] Check Spelling

Links to a user-supplied spelling checker. If a spelling checker has been installed on your workstation and linked to WinCALIS, selecting this option will start the spelling checker. The WinCALIS Author Workstation Setup package includes a shareware Windows spelling checker program, WinSpell 3.0®, which may be freely distributed.

[Tools] Print File (TrueType)

Selecting this item launches the WinCALIS TrueType Multilingual Printer Program.

[Tools] Print File (Bitmap)

Selecting this item launches the WinCALIS Graphic Multilingual Printer Program.

[Tools] Tools Menu Readme

Selecting this item displays information about the Tools menu and some of its functions.

Help Menu (WinCALIS)

[Help] General

Provides the same general help as the WinCALIS opening screen on broad topics ranging from “Choosing a lesson” to “Working efficiently.” Use the index and help system commands for searching for more specific WinCALIS topics and commands.

[Help] Keyboard

By selecting this option, a window containing a graphic display of the current keyboard setup (i.e. which character is mapped to which key) will be displayed of the language in which one is working and also the procedure of using keystroke combinations to form diacritics. Notice that, while typing, pressing Shift or Ctrl shows the shifted state or state with Ctrl pressed of the virtual keyboard. Characters can be inserted in text directly from virtual keyboard with left (primary) mouse click; the effect of overstrike accents on other (accented) characters is shown with a right (secondary) mouse click.

[Help] Author Help:

This option gives access to a customized Help file for use with the Windows 3.1 Help system for a particular exercise or group of exercises, if one has been supplied by the author of the exercise.

[Help] About...

About... shows the current version and date of your copy of WinCALIS. It also displays the serial number of the package and the name it is registered to.

Accents and Diacritics

In order to produce diacritics (i.e. a modification of a character, such as an accent) and to accommodate more characters than the standard English keyboard normally allows, combinations of keys are sometimes used. In order to modify characters with diacritics, one must gain a familiarity with the keyboard layout.

The Overstrike Key = F2

The “overstrike key” F2 is used to make accents and diacritics. It functions much like the Ctrl, Shift, and Alt keys on the standard keyboard. Pressing F2, or selecting the menu item “Add Accent,” will highlight the letter to the left of the insertion point (with blinking caret) and the next letter pressed will overstrike the highlighted letter as a diacritical modifying accent (grave, acute, circumflex, tilde, diaeresis, etc.).

Accent Keys = Pale Blue

Each diacritic is associated with a particular character on the keyboard, appearing in pale blue in the Keyboard Help virtual keyboard display. An individual diacritic may modify several characters, such as a single accent working with several vowels. In order to display the character set associated with a particular diacritic, right-mouse-click on one of the light blue accent keys or press the letter on the keyboard.

Note: Because accents like the circumflex (^), the tilde (~), and the double quote ("), used to insert the umlaut, are shifted keys, Shift must be pressed along with the accent key.

The accent key will turn red and the associated characters will appear in green. In some cases there are more diacritics than displayed on the original Help keyboard. Clicking on or pressing the Shift key will display additional accents and diacritics.

Thus to see the full set of characters modified by the tilde (~), you must first press Shift + ` , which will display the upper-case characters (shifted keys) modified by the tilde, then click on or press Shift again to display the lower-case characters (unshifted keys) modified by the tilde.



Figure 3: Putting the Keys Together

After the overstrike key, accent keys, and their associated characters have been identified, the three may be used in conjunction to create an accented character. The general procedure is to type the character itself, the overstrike key, and the accent key in that order. Notice when the overstrike key has been pressed, the character appears in reverse video, ready to be modified with the diacritic. For example:

Using the English/Western Europe keyboard (which supports all Roman languages) to create an ê:

1. Type the lower case "e"
2. Press the "overstrike key" F2
3. Press Shift+6

In order to create capital letters simply type the capital letter and then proceed as described above.

Note: You may insert characters directly into your text from the virtual keyboard by clicking on any character. Use the right (secondary) mouse button on a light blue key to see all of the characters that are affected by adding that accent/diacritic.

At this point, you are prepared to tackle any WinCALIS lesson. Lessons may vary in language, content, style, and pedagogical technique, but you are now equipped with the fundamental user knowledge needed to cope with most lessons. If you need more help or practice, go through the demonstration again or any of the other lessons that come on the WinCALIS disks. When you are comfortable with WinCALIS, you should move on to the real focus of this manual, which is authoring. Like any form of creation, we believe that you will find authoring quite rewarding. But unlike most forms of creation, you will find it quite easy when you use the WinCALIS authoring tool.

Chapter 2: Authoring Basics

Before You Begin Authoring

After seeing what WinCALIS can do, you are now ready to author your own lessons. Learning to use WinCALIS Author will be like learning to drive a car, only a lot more rewarding and a lot less stressful. There are some fundamentals to be learned, but after you have them down, you won't even need to think about them, and you will have the ability to take your students to new heights with your lessons.

As with driving, the best way to learn is by doing. Therefore, we will provide some hands-on experience so you can get a feel for WinCALIS Author and its potential. You will create your first script by following our step-by-step instructions. If you are new to the WinCALIS system, you may not understand everything the instructions tell you to do. That's perfectly fine; we'll explain all the details later, after you've created your first lesson and have seen WinCALIS Author in a practically applied context.

To preface your first lesson, you should know a few simple things about WinCALIS Author. First, WinCALIS pierces the language barrier with Unicode™, a character coding standard containing all of the world's character sets, making it possible for WinCALIS authors to develop lessons in any language of the world. Second, for every window you have seen operating in WinCALIS, there is a corresponding window in WinCALIS Author which is used to implement its WinCALIS counterpart. For example, the Presentation (Text) Window in any WinCALIS lesson is created by a Presentation (Text) Window that the author uses in WinCALIS Author. And thirdly, for every question in a lesson, the author must both create the question itself and tell WinCALIS Author the right answer. Optionally, the author may include more than one right answer, anticipate wrong answers, and provide feedback for any answer, right or wrong. These principles will be explained in much greater detail after you create your first script. In the next chapter, you will learn how to customize your lessons linguistically and aesthetically. But for now, you'll stick to the basics.

Lastly, we highly recommend that you create a subdirectory on your computer under the standard WINCALIS\SCRIPTS directory, which was automatically created when you installed the program (if you accepted the default--otherwise substitute the directory path you requested). You will use this subdirectory to store your lessons. “\WINCALIS\SCRIPTS\WORKING” would be a good choice.

Now, let's begin!

Refer to your Windows manual on how to create subdirectories.

Your First Script

After you have started Windows, move to the WinCALIS Pangaea group, then select and run WinCALIS Author. (To run the program, either double-click on the Author icon, or use the arrow keys until the Author icon is highlighted, and then press Enter.) The opening screen will appear. There are six windows in the Author program: the Lesson Description Window, the Presentation (Text) Window, the Popup (Directions) Window, the Interactive (Question) Window, the Pass Through Window, and the Scratchpad Window. The Presentation (Text) Window, Interactive (Question) Window, and Pass Through Window should be showing, with the other windows minimized at the bottom of the screen. Any one of the windows on the screen can be made active by simply moving the mouse pointer to the

desired window and clicking (if you have a choice, it's the left mouse button). You may also go to the Window menu and select the window you would like to activate from the list at the bottom. The active window on the screen is denoted by a change in the hue or intensity of the title bar.

Now that we are situated, we will begin our step-by-step creation of your first script. Here are the steps that will be detailed in this tutorial.

1. *Giving Directions*
2. *Creating a Presentation (Text) Window*
3. *Producing a Multiple Choice Question*
4. *Producing a True/False, True Question and a True/False, False Question*
5. *Producing a Short Answer Question*
6. *Producing a Cloze Question*
7. *Producing a Tutorial Window*
8. *Saving the Lesson*

We're ready to start your first lesson!

1. Giving Directions

The first thing you will learn how to do is how to give directions in a lesson. To begin, click on the Popup (Directions) Window in order to make the window active. You should notice the title bar change hue. Maximize the Popup (Directions) Window by clicking on the Maximize button in the window's upper right-hand corner.

It is a good idea to place one blank line before all typed text to serve as a top margin. At the prompt, after a blank line, type:

Read the following text carefully and answer the questions.

Restore the Popup (Directions) Window by clicking on the Restore button in that window's upper right-hand corner. (Then minimize the Popup (Directions) Window, if you wish, by clicking on the Minimize button in the Popup (Directions) Window's upper right-hand corner. You may have to double-click on the title bar to access the Minimize button.)

Giving directions is as simple as that.

You will later add more text to the Popup (Directions) Window (toward Producing a Tutorial Window).

2. Creating a Presentation (Text) Window

Next we will learn how to create a Presentation (Text) Window. Make the Presentation (Text) Window active by clicking in the window. Maximize the window, then type the following in the Presentation (Text) Window, after one blank line. Keep in mind that words appearing within brackets [] are instructions for you. Do not type them in your text. If you are using the on-line Help version of the tutorial, widen the window in order to see the correct line formatting. (Alternatively, you may import a previously typed copy of the text into the Presentation (Text) Window by clicking on File, Document file, Open, and selecting the file

"MANET-A.TXT." A more extensively formatted version of the same text is in the file "MANET-B.TXT.")

[3 tabs] *The European, 22-23 May 1993, Stephanie Theobald*

<TAB:8><GRAPHIC:manet.bmp>

[3 tabs] *Turmoil at the*

[3 tabs] *grass roots level*

[picture will appear in this space]

If Manet's naked woman from Déjeuner sur l'herbe tried to eat her lunch on the grass in 1993 France, she would face a fine of up to Ffr 2,000 (ECU 304) -- and not because of her lack of clothes.

The battle between Paris park-keepers chasing people off the lawns and infuriated taxpayers demanding their rights to walk on the grass is [1 tab] Edouard Manet - Déjeuner sur l'herbe - 1863 heating up. The Paris Park and Public Garden Authority, in charge of the capital's 400 parks, fears that tensions are about to reach the boiling point.

The authority is receiving 50 letters every week from Parisians complaining that in spite of the Ffr 825 million of taxpayers' money invested this year in the public gardens, they are still being forced to sit on hard wooden benches. The authority's communications director, Dominique Maurel said walking on the grass had always been illegal. But he added, "It does seem slightly ridiculous now that we're not all going round in bowler hats that people are not allowed to sprawl around in public green spaces."

Some of Paris's mayors have taken steps to call a halt to this unrelaxed atmosphere. The mayor of the 15th arrondissement has opened up a park where everyone is allowed to walk on the grass.

But the mayor's office for the first arrondissement, which has created a grassy area for children under 12, warned that it was just an experiment. And said the office was aware of the old Parisian ruse of pretending to be a parent for an hour in order to gain access to tempting areas of green grass on hot summer days.

An insurance salesman taking his lunch break in the chic Parc Monceau doubted Parisians will accept the "intolerable situation much longer." "Look at the parks of Versailles and Fontainebleu. In the heyday of the 17th century even they were more relaxed. Remember the fun Marie Antoinette had in the bushes," he said.

Stephanie Theobald

Now restore the Presentation (Text) Window.

Note: Because the word wrap capability is disabled as the default in the editor, make sure that you press Enter when you wish to start a new line.

3. Producing a Multiple Choice Question

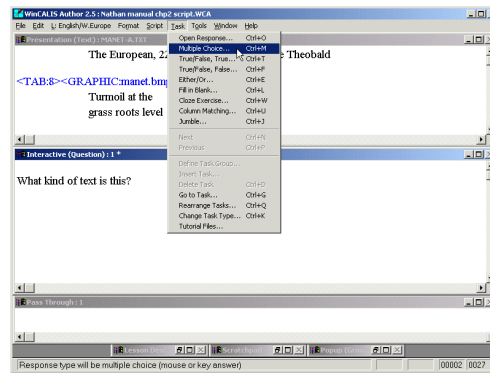


Figure 4: Selecting Multiple Choice Question Type

Now, it's time to learn how to make a multiple choice question. Using the mouse, click in the Interactive (Question) Window to make it active. The cursor will appear at the beginning of the Interactive (Question) Window, ready for you to start typing your questions. Type:

What kind of text is this?

Click on the Task menu in the main menu bar at the top of the screen, and select the Multiple Choice menu item. You will be presented with a Multiple Choice dialog box. On the left-hand side of the Multiple Choice dialog box you will notice four Response Types. The default is noted by a darkened radio button. The default Response Type for the Anticipated user response is Correct/Continue.

The cursor will be blinking in the Anticipated user response or choice text box. Type:

newspaper article

You have just “defined” a response that WinCALIS will recognize as satisfactory, should the user choose it as an answer, and the user will continue to the next task (The task flow will Continue).

Having finished with the first anticipated response, click on the Next Response button on the right-hand side of the dialog box. WinCALIS Author will clear the Anticipated response text box in order to accommodate the next choice. The cursor will again appear at the top left-hand corner of the Anticipated response text box. For the rest of the Multiple Choice entries we will anticipate the Response Types as Incorrect/Remain Here. Use the mouse to select Response Type Incorrect/Remain Here and type:

advertisement

in the Anticipated response text box. WinCALIS will recognize the answer as incorrect, and the task will be presented to the user again (The task flow will Remain Here).

Using the Tab key or mouse, go to the Computer's response/your feedback (optional) text box. Remembering to

press Enter when you wish to start a new line, as the word wrap capability is disabled, type:

An advertisement usually has a much more attractive layout and focuses on a product or service to be sold.

Remember: Providing feedback is always optional.

The message you typed in the Feedback text box will appear in WinCALIS, should the user choose “advertisement” as a response to the question.

Click on the Next Response button and repeat the process for the following, incorrect anticipated responses:

telephone directory

and as feedback:

A telephone directory is a long list of telephone numbers together with people's names and addresses.

Then:

page from a manual

with the feedback:

A manual would have less full text and more short, clear instructions in a strictly outlined format.

Click the done button. Note that clicking the Done button moved you on to the next task. You are now in Task 2, as shown by the “2” in the active window.

Since it is a good idea to save your work every 15 minutes or so when using a computer, now is a good time to save your lesson. In the File menu, under the Script database (*.wca) heading, select the Save As... menu item. You will be presented with a Save Author Script Database dialog box, which will ask you to provide a name and directory for the file. If you wish, you may choose a new directory for the file to be saved in by selecting one of the folders shown in the dialog box or by selecting one of the folders found in the pull-down menu next to the Save in heading, at the top of the dialog box. Once you have chosen a directory in which to save your script, type in a name for your script in the File name box. WinCALIS Author will attach its own suffix .WCA to any name you provide, so you don't need to give it a suffix yourself. We suggest that you name your file with a name indicative of its content. In this case, the file name “firstles” works well.

Click on OK when WinCALIS Author tells you “Script (.CAL file) generation successful.” This means that you have saved your Author database file “FIRSTLES.WCA” to disk, and from it Author has created a WinCALIS script file, “FIRSTLES.CAL.” You will be returned to the WinCALIS Author screen.

Note: The following Response Types are available:

Correct / Continue	For all possible correct responses; next task will be presented
Incorrect / Remain here	For all possible incorrect responses; task will be re-executed
Neutral (Re-execute)	For responses that will count as neither correct nor incorrect
Neutral (No Re-execute)	This option is used for audio, video, and pop-up windows

4. Producing a True/False, True Question and a True/False, False Question

True/False, True

Now you will learn how to produce the two types of True/False Questions: True/False, True and True/False, False. It's easier than the names sound. In the Presentation (Text) Window, type the following statement:

Walking and sprawling around in public green space in Paris is prohibited.

Click on the Task menu and select the True/False, True menu item. You will be presented with a True/False dialog box. Notice that the Response Type is set to Correct/continue and the word True appears in the Anticipated user response or choice text box.

In order to give helpful feedback should the student click on the Incorrect/Remain Here answer, which in this case would be False, click on the Next Response button on the right-hand side of the dialog box, then tab to or click into the Computer's response/your feedback (optional) text box and type an appropriate message.

Once you are finished, click on Done.

True/False, False

In the new Interactive (Question) Window type the following (For an é type e then F2 then '. See Accents and Diacritics in Chapter 1 for further information):

Morals have changed in France. Because she is naked, the woman from Manet's Déjeuner sur l'herbe would get a fine when lunching on the grass in 1993 France.

Click on the Task menu and select the menu item True/False, False. Notice that the Response Type is set to Correct/Continue and the word False appears in the Anticipated response text box. Again click on Done.

5. Producing a Short Answer Question

In this step, you will learn how to create a short answer question. In the new Interactive (Question) Window type:

What do Parisian taxpayers demand from the authorities?

Click on the Task menu in the main menu bar and select the Open Response menu item. The Open Response (Short Answer) dialog box will appear. The Response Type will be set to

Correct/Continue, and the cursor will be blinking in the Anticipated user response text box. Type:

They demand the right to walk on the grass.

Having finished with the first anticipated response, click on the Next Response button. In the cleared Anticipated response text box, type:

&{walk|sprawl}&{grass|green|park}&*

Note: The wild card character | is a 'pipe' or 'bar' (not a slash). It can be found near the backspace key on some keyboards.

Wild Card Characters

- ^ Matches any single character
- [] Matches any *one* of the enclosed characters
- [-] Matches any *one* character in sequence
- * Matches any sequence of characters within a word
- & Matches any sequence of characters, including spaces
- | Matches alternatives separated by vertical bar
- { | } Matches any one of the enclosed alternatives separated by vertical bars

Note: Wild Card characters are explained in detail in Chapter 4, Customizing and Communicating.

Once more click on the Next Response button. Now you will anticipate an Incorrect/Remain Here response. Select Incorrect/Remain Here from the Response Type options, and type in the Anticipated response text box:

sit on hard wooden benches

Go to the Feedback text box and type:

Please read the second paragraph again.

When you are finished, click Done to go to a new Interactive (Question) Window.

6. Producing a Cloze Question

Now you will learn how to make a cloze question. In the new Interactive (Question) Window, type the following text. If you are using the on-line Help version of the tutorial, widen the window in order to see the correct line formatting. (Alternatively, you may import a previously typed copy of the text into the Interactive (Question) Window by clicking on File, Document file, Open, and selecting the file "CLOZTEXT.")

*A conflict between Parisians and authorities has arisen.
Citizens are demanding their rights to walk freely on the
grass in the parks whose maintenance they are paying for.*

*Though the park authority's communications director
seems to agree with their demands, he said walking on the*

grass had always been illegal.

In spite of this, mayors of two Parisian districts opened green areas to the public as an experiment.

Press Shift + ? or click on Clue button for clues

Go to the Task menu and select the Cloze Exercise menu item. A Cloze (Fill in blank) dialog box appears with the text that was previously entered in the Interactive (Question) Window.

Marking Text in a Cloze Exercise

With the Cloze (Fill in blank) dialog box active, highlight the word *Parisians* in the Cloze (Fill in blank) text box by double-clicking the word. With the word highlighted, click on the Mark button in the right-hand side of the dialog box. An information box will pop up, warning you that you cannot edit your text once you are in mark phase. Click on Yes.

Repeat the steps described above for the word *authorities*.

Additional Anticipated Responses

Having finished with the first set of anticipated responses, you may want to anticipate additional responses, whether Correct/Continue or Incorrect/Remain Here, for the individual words chosen in the cloze task. In order to accommodate additional predictable responses, you must do the following:

a- First highlight the word(s) *Parisians* in the Cloze (Fill in blank) text box. With *Parisians* highlighted, select the More button.

b- A More Answers/Feedback dialog box will open, with the word *Parisians* in view.

c- Click on the Next button, located at the right of the text box. WinCALIS will clear the Anticipated user response text box in order to make room for additional responses.

d- Now go to the Response Type option box and choose Incorrect/Remain Here. In the Anticipated response text box, type:

citizens

Using the Tab or mouse, go to the Computer's response/your feedback (optional) text box and enter:

Citizens of which city?

e- Click on the Done button in the More Answers dialog box to save the contents of the dialog box as part of the cloze task.

f- Repeat steps **a** through **e** for the response *authorities*:

Enter *park-keepers* and *mayors* as additional Correct/Continue answers, and *gardeners* as the anticipated Incorrect/Remain Here answer.

Adding a Contextual Clue

In the Cloze (Fill in blank) text box, mark the word *demands* by double-clicking on it and then clicking on the Mark button.

In order to create a contextual clue from the phrase *demanding their rights to walk freely*, highlight the phrase by clicking and dragging, and click on Clue.

Adding a Noncontextual Clue

First click in the Noncontextual Clue text box at the bottom of the Cloze (Fill in blank) dialog box. Type:

"a test"

Then highlight *experiment* by double-clicking the word in the text and mark it by clicking on Mark.

Click on Done to save the entire contents of the Cloze (Fill in blank) dialog box for Task 5.

7. Producing a Tutorial Window

In our last step, we will learn how to produce a Tutorial Window. Tutorial Windows that can be accessed by the student are created in the Scratchpad. In this Tutorial Window we will include some definitions of words used in the text and emphasize the headings and some other words using color and other screen attributes, such as bolding and italics.

Make the Scratchpad active by clicking on the Scratchpad icon, if visible, or by selecting the Scratchpad menu item from the Window menu. Maximize the Scratchpad Window by clicking on the maximize button in the top right corner of the Scratchpad.

Now in the Scratchpad type the following text, adding the formatting commands where needed. If you are using the on-line Help version of the tutorial, widen the window in order to see the correct line formatting. (Alternatively, you may import a previously typed copy of the text into the Scratchpad by clicking on File, Document file, Open, and selecting the file "VOCAB-B.TXT.")

To add color to each heading ("Turmoil," "grass roots level," etc.) highlight the word(s) with the mouse, then click on the Format menu and select the Color

menu item under the Selection heading. Then click on the color Dark Blue to select it. This will insert the CALIS commands <DKBLUE> surrounding the highlighted text, which will display the enclosed text in dark blue color when the exercise is run in WinCALIS.

To make bold the name Manet, similarly highlight the word Manet with the mouse, then click on the Format menu and select the Bold menu item under the Selection heading. This will insert the CALIS commands <BO> surrounding the highlighted text, which will display the enclosed text as bold type when the exercise is run in WinCALIS.

To add italics to a whole section of text, such as the list of Manet's paintings, you will set the screen attributes to italics, then back to normal. To do this type the commands <SET ATTR IT> and <SET ATTR NORMAL> before and after the text to be italicized, so that it looks like this:

(<SET ATTR IT>Le déjeuner sur l'herbe; Olympia; Le balcon, Le fifre; l'Exécution de Maximilien<SET ATTR NORMAL>)

[5 tabs] *Turmoil*

tur-moil n.

1. *a state of great commotion, confusion, or disturbance; tumult; agitation; disquiet.*
2. *Obs. hard labor; toil.*

[5 tabs] *grass roots level*

n. (used with a sing. or pl. v.)

1. *ordinary citizens, esp. as contrasted with the leadership or elite.*
2. *the agricultural and rural areas of a country.*
3. *the people inhabiting these areas, esp. as a political, social, or economic group.*
4. *the origin or basis of something.*

[5 tabs] *Manet*

Manet, Edouard, French painter, born in Paris (1832-1883).

Often inspired by the Classic Masters, and in particular by the Spaniards of the Age of Enlightenment. He inspired several leading Impressionists.

(Le déjeuner sur l'herbe; Olympia; Le Balcon, Le Fifre; l'Exécution de Maximilien)

[5 tabs] *Déjeuner sur l'herbe:*

Famous painting by the French painter Edouard Manet, (1832-1883) (See image.)

[5 tabs] *Ecu*

ECU n. pl. (ECU's, ECUs)

1. *a money of account of the European Economic Community.*

Save the Tutorial you just created. In the File menu, under the Document file heading, select the Save As... menu item. A dialog box will pop up. In the File name box, name the document *vocab.txt* (Do not name it *VOCAB-B.TXT*.) Click on Save.

Now you will place the Tutorial Window in your script, so that it will be available from the beginning of your lesson. First, minimize the Scratchpad Window in Task 6. Now select the Previous menu item from the Task menu (or use the shortcut Ctrl+P or the '-' key on the number keypad) until you return to Task 1.

In the first task, activate and restore the Popup (Directions) Window. Scroll back to the left using the scroll bar at the bottom of the window until all the text is showing. Place your cursor two lines below the existing directions and type:

*For vocabulary review press Shift+F2
anytime after reading these directions.*

Move to the Task menu and select the Tutorial Files menu item. You will be presented with the Tutorial Files text box.

Click in or Tab to the Shift+F2 window under the Text file name heading. Type the name of the file you just created (*vocab.txt*).

Click in or Tab to the Window title box and type:

Vocabulary.

When you have finished, click on OK.

8. Saving the Lesson

At this point, you have finished making a short, five-question lesson. Now all you have to do is save and run it. In the File menu, under the Script database (*.wca) heading, select the Save As... menu item. You will be presented with a Save Author Script Database dialog box, which will ask you to provide a name and directory for the file. If you wish, you may choose a new directory for the file to be saved in by selecting one of the folders shown in the dialog box or by selecting one of the folders found in the pull-down menu next to the Save in heading, at the top of the dialog box. Once you have chosen a directory in which to save your script, type in a name for your script in the File name box. WinCALIS Author will attach its own suffix .WCA to any name you provide, so you don't need to give it a suffix yourself. We suggest that you name your file with a name indicative of its content. In this case, the file name "firstles" works well.

Click on OK when WinCALIS Author tells you "Script (.CAL file) generation successful." This means that you have saved your Author database file "FIRSTLES.WCA" to disk, and from it Author has created a WinCALIS script file, "FIRSTLES.CAL." You will be returned to the WinCALIS Author screen.

Running a WinCALIS Lesson

1. Return to Windows either by switching between applications with the Alt+Tab key, or exiting Author and starting up WinCALIS (it's OK to exit WinCALIS Author by clicking on the File menu and selecting Exit, but after you run the lesson in

WinCALIS, you may want to go back and modify the lesson). Thus, you can save time by leaving Author running in the background and using the Alt+Tab key to go directly to the Program Manager and clicking on the WinCALIS icon). Click on the Options menu in the menu bar. Select the menu item "Text Conversion [*.CAL]." This tells WinCALIS what kind of file to look for.

2. If the directory name displayed above the Lesson Window in the dropdown box after the heading "Look in:" does not indicate the directory in which you stored FIRSTLES, then select that directory by clicking on the down arrow in the dropdown box or on a folder icon and navigate through the directories. When you have the right directory, you should see the name of the lesson (FIRSTLES.CAL) alphabetized somewhere in the list in the Lesson Window. Highlight the lesson name by clicking on it or using the arrow keys and click on the run button to start the lesson.
3. The lesson you created will appear right before your very eyes! There are several things to keep in mind while you execute this first lesson. Remember that until you learn some more tricks to authoring, your responses must match the original exactly to be correct. Try entering a wrong answer to see how WinCALIS reacts. Notice how WinCALIS signals you to continue on to each question. Remember that you (as the student) can use the function keys as shortcut keys to perform many tasks faster. (See the chart below.) When you are finished with the lesson, you will be told how you did.

WinCALIS Shortcut Keys

<i>To</i>	<i>Press</i>
Get on-line help	F1
Add Accent	F2
Switch between two currently used languages	F3
Look Up in Dictionary	F5
Give Up	F6
Show Directions	F9
Play Audio	F11
Play Video	F12

If you think you have the hang of these fundamentals, feel free to reopen the script in WinCALIS Author, tinker around, then run the script again in WinCALIS. You might create some more questions, change the feedback, or, if you are confident enough, make an entirely new lesson with your own text, questions, and feedback.

Overview of WinCALIS Author

If you have authored a real WinCALIS lesson (Your First Script), you should have a feel for how WinCALIS Author works. At this point, it's good to keep in mind that almost everything you see in a WinCALIS lesson, as in the demonstration, is brought to life by a corresponding action in WinCALIS

Author. We will now provide some background information to accompany the practical demonstration you have finished. In these sections we will explore the details of implementing lessons in Author. As a teacher, you're familiar with the instructional process most formal education follows. To summarize you acquaint the students with the learning objective, give them directions, provide the information to be learned, question the students on that information and then provide appropriate feedback. WinCALIS Author allows for easy implementation of this process with the refreshing and powerful aid of a computer. Each of the WinCALIS Author windows is designed to implement an element of that five-step process.

The Lesson Description (Comment) Window tells the user what the learning objectives will be and what to expect in the lesson in terms of subject matter. The Popup (Directions) Window tells the student *how* to go about fulfilling those objectives. The Presentation (Text) Window provides the actual substance of the lesson, be it a short story, grammar review, or newspaper advertisement. And lastly, the Interactive (Question) Window asks the student about the learned information and provides appropriate feedback, encouragement or clues just as if you were physically present helping the student. If the user has trouble with any of the areas covered by the lesson, you have the option of providing pop-up Tutorial screens for review. This will be explained further in the section Customizing and Communicating. The entire system is simple and practical.

When you are designing your lessons, it will be helpful to keep in mind this systematic approach to design. Naturally, all the parts of a lesson are interrelated, but planning in a particular sequence will facilitate your work.

WinCALIS Author and Unicode

WinCALIS Author editor is a relatively simple text processor, but a powerful multilingual editing system. By incorporating support for Unicode™ the editor combines all of the world's alphabets into one program, making WinCALIS truly multilingual. The following will offer a quick overview of some of the basic rules and methodologies you must know in order to understand how the WinCALIS Author editor works.

The WinCALIS authoring editor is geared towards foreign language learning. It does this by incorporating support from Unicode, a standard code that can map the combination of all the world's character sets onto your standard keyboard. Using two bytes for each character representation, rather than one which the standard computer ASCII format uses, 65,536 unique characters can be represented, allowing WinCALIS to be multilingual and international. The aim of WinCALIS was fourfold. The first was to offer completeness in all of its languages. Unicode provides this by allowing WinCALIS to support up to 65,000 individual characters. The second aim was simplicity, such that every unique character is represented with no duplication. The third was correctness to ensure that every character is a real character as defined by experts. Finally, we wanted to ensure fidelity so that no information would be lost when converting text to or from existing character sets. In order to run earlier written lesson scripts, WinCALIS has a built-in transparent translation facility, but it is also prepared for the future. Windows NT, Microsoft's first 32-bit operating system, also can use Unicode internally.

See Also: Unicode Character Selection Dialog Box

The Editor

The WinCALIS Author Editor is a simple word-processing application that you can use to create and print scripts under WinCALIS Author. To write scripts, you need to know how to type, edit, and format text in the WinCALIS Author windows, save the file as either a Script file or a Document file, and print the Document file document on your printer.

All the editing windows in WinCALIS Author share the following items:

Control Menu

The Control-menu box is in the upper-left corner of each window. The Control menu is most useful if you use your keyboard to work with Windows. By using Control menu commands, you can restore, move, size, maximize, minimize, and switch to other applications. (If you use a mouse, you can perform these tasks by clicking and dragging.)

Title Bar

The title bar shows the names of the editor windows that are in the Restore form along with the task number you are currently working on.

Maximize/Minimize Buttons

Using a mouse, you can click the Maximize button (up arrow) in the upper right-hand corner to enlarge the active window to fill the entire screen, or you can click the Minimize button (down arrow) next to it to reduce the window to an icon.

Restore Button

The Restore button takes the place of the Maximize button once a window has been enlarged. The Restore button contains both an up arrow and a down arrow. If you click the Restore button, the window will return to its previous size.

Vertical/Horizontal Scroll Bar

These can be used to move parts of a document into view when it doesn't fit into a window.

Window Border/Window Corner

The window border is the outside of the window. By moving the mouse pointer to the window border, you will notice the mouse pointer changes from a vertical bar to one with horizontal or vertical double arrows. You may shorten or lengthen each side of a window by holding down the mouse button and dragging the border when the mouse pointer shows the horizontal or vertical double arrows.

The small box at the corner of each window, called the window corner, also may be used to adjust the size of the window. It works like the window border, but adjusts the sides of the windows at the same time.

Insertion Point

The insertion point is the location in which the text will appear when you start typing, where there is a blinking "caret."

Mouse Pointer

The mouse pointer changes with the movement of the mouse. When you select Context-Sensitive Help from the Help menu, or press Ctrl+F1, the mouse pointer changes into a Help pointer. You may click the Help pointer on any menu item or CALIS command, special response or option keyword to get on-line help about it.

For a detailed description of the main menu bar, see the section WinCALIS Author Menus.

WinCALIS Editor Limits

Note: Authors should be aware of the following size limitations, which affect the editing and display of text in WinCALIS and WinCALIS Author windows at all times:

- A maximum of 6000 lines of text is permitted in each window in WinCALIS Author (Scratchpad, Interactive (Question) Window, Presentation (Text) Window, etc.). This is the equivalent of 200-300 pages of double-spaced text. (Fewer than 6000 lines may be available if there are limited Windows system resources, or when WinCALIS and WinCALIS Author are operating at the same time.)

Tip: The TrueType Printer program provides a handy place for opening and viewing WinCALIS Unicode files of virtually unlimited length. And since the file is opened in read-only mode, it has some of the speed and safety of other file viewers for DOS and Windows, like “LIST” or Xtree. You can edit a very long file in segments of less than 6000 lines, with all but the first segment saved as type “Unicode/No Header.” Then use the DOS command “copy /b”file1 + file2 + file3 hugefile” to concatenate the separate segments into one big file. This is the procedure used for creating the large on-line WinCALIS dictionaries.

- A maximum of 1000 lines of text is permitted at a single time in any single WinCALIS window (Question, Directions, Text, Lesson Description, Tutorial windows, etc.).
- For word-wrapped text in Author (with soft returns), there is a limit of approximately 5000 characters per paragraph (text between hard returns). This limit primarily affects general text editing, in which Word Wrap has been enabled. For ordinary lesson authoring, in which each line ends with a hard return, this limit has little effect. However, the following limit should be observed.
- When producing WinCALIS lessons in Author (as opposed to general text editing), a maximum of 240 characters per line is permitted, including non-displaying text such as CALIS enclosed in angle brackets. Author will truncate lines longer than 240 characters when generating the *.CAL script. If there are numerous non-displaying CALIS commands in a line which is approaching 240 characters in length, be sure to use the abbreviated forms of commands (e.g., <G> instead of <GREEN>, <IT> instead of <ITALIC>, <GR:filename> instead of <GRAPHIC: filename.bmp>, etc.).
- Author will handle, but not display properly, the ends of very long lines (greater than about 260 “Medium” 21-pixels-high characters in length. Such lines can be edited by breaking them into two or more shorter lines, then performing needed editing, then recombining the lines by deleting the hard returns. Alternatively, Word Wrap can be selected from the Edit menu, and the long line will be temporarily wrapped, if it is edited (e.g., insert a space character, then delete it) or the file is opened from disk with Word Wrap selected.
- For cloze texts in Author, there is a limit of 64K per text (roughly 32,000 characters).
- When mixing right-to-left text (Arabic or Hebrew scripts) and left-to-right text, a maximum of 13 changes of direction is permitted per line (between hard returns). This limit can easily be exceeded if numerous CALIS commands are used in text, since CALIS commands are left-to-right in directionality. A typical problem case is that of marking text, when “marks” such as “<M:a1>” are inserted in right-to-left text to identify the beginning and ending of “hotspot” hyperlinks. A rule-of-thumb to go by is that a maximum of three marked words or phrases may be used successfully in a single line--four, if one of the marks is at the beginning of a line (which will not require a change of direction, since the CALIS statement marker at

the beginning of the line is already left-to-right). Usually, line lengths can be edited so as to avoid inconvenience.

- For cloze texts in Author, there is a limit of 64K per text (roughly 32,000 characters).
- Task labels must be no longer than 15 characters. Any characters beyond 15 will be ignored by WinCALIS. Thus the following two task labels will look the same to WinCALIS:
#verylongtasklabel1
#verylongtasklabel2
- Multimedia clip names must be no longer than 19 characters, with no internal spaces.
- No more than 50 instances of importing files with the <IMPORT tutfile> command can be processed by WinCALIS when it runs a *.CAL script file and converts it into a *.WCL file. This is due to the way in which WinCALIS checks time-date stamps to see whether it needs to convert new import files into *.WCI files during the script conversion process. There is no limit on the number of *.WCI import files which WinCALIS can handle when directly executing ordinary *.WCL script files. See <IMPORT>ing Tutorial Files.
- The student answer in WinCALIS should be no longer than 99 characters (55 character between spaces, since the buffer which holds the student answer is 99 characters long. An answer longer than 99 characters is likely to cause erratic behavior in WinCALIS).

The WinCALIS Author Environment

WinCALIS Author arranges its windows and information like any other Windows application. Menus pull down, windows are maximized (opened full-screen) and minimized (reduced to icons), sized, and moved, all with the click of the mouse. Under the Window menu, options allow you to “tile” your windows, i.e. have them all displayed at once like tiles in a floor, or “cascade” them, i.e. have them overlap like folders with the most recently used ones on top. At any time you may also “arrange” your minimized window icons with the Arrange Icons menu option.

The active window, that is, the one in which you are able to enter information at any given moment, will be denoted by a change in intensity or hue of its title bar. To activate a window, you may click in it once or select it under the Window menu. Moving from window to window is that simple.

WinCALIS Author Windows

Lesson Description Window

The Lesson Description (Comment) Window allows the author to enter the text that will be displayed when a lesson first starts. This text appears in the Lesson Description Window when a particular lesson is highlighted on the WinCALIS opening screen. The Lesson Description Window normally describes the lesson and can provide a focus for the lesson before the user begins.

Note: The Author Workstation version of WinCALIS has a “Text Conversion” mode which is not available on the ordinary Student Workstation. The contents of the Lesson Description window do not appear in “Text Conversion” mode. In order to view what is displayed to the student in the Lesson Description Window, the “No Conversion (*.WCL)” option should be selected from the WinCALIS opening screen Options menu.

Popup (Directions) Window

The Popup (Directions) Window gives information to the user such as instructions for completing the lesson, method of scoring, whether “giving up” is allowed, and other formatting aspects. Be as clear as

possible in any directions you give. It's always best to assume that your users have little computer experience, especially in early lessons. WinCALIS is easy to use for both students and authors, nonetheless beginners are often nervous in front of computers.

Presentation (Text) Window

The Presentation (Text) Window contains the context for questions in most lessons. Text is usually synthesized by the author or taken from novels, magazines or text books. Certain types of text selections lend themselves to particular question formats, but almost any question may be asked of any text selection. The most important thing about the text is to make it as interesting as possible. It's hard to make learning gerunds interesting, but one might invent a fantastic story to frame the grammar lesson.

For some lessons, such as strict tests of grammar, vocabulary, or previous reading material, a text selection may be unnecessary or undesired. In these instances, it is acceptable to forego use of the Presentation (Text) Window.

Interactive (Question) Window (With Answers and Feedback)

The Interactive (Question) Window contains the pool of questions to be administered to the user after reading the directions and the text selection. Before you type any questions, you must decide what question format(s) you want to employ. This depends primarily on the nature of the text selection, your particular pedagogical goal, and the ability and experience of the users. For some text selections, you might want to use true/false and multiple choice, while for others you might want to give short answer comprehension questions only.

Only one question will fit in the Interactive (Question) Window at one time, but it is easy to scroll through the list of questions. You might think of the questions you develop as being written on index cards, and scrolling as a way of flipping through the cards. The types of questions available are: short answer, multiple choice, true/false, and cloze--all of which are explained in the section Question and Feedback Development.

Once you have devised a question, you must tell WinCALIS the answer. After the user enters a response to a question, WinCALIS checks the list of responses you have provided and gives the user appropriate feedback, which you may also provide. Every question may have multiple Correct/Continue answers, but must have at least one Correct/Continue response. Authors can anticipate variations on each response, such as capitalization, punctuation, spelling, and word order. WinCALIS has several features which facilitate the analysis of a broad range of responses.

As with classroom teaching, the feedback you provide to questions is often just as important to the learning process as the questions themselves. Feedback can lead a student to the right answer, and discourage him or her from giving up too soon. The most valuable learning often comes from the search for the answer!

Scratchpad Window

The WinCALIS Author Scratchpad Window is a Unicode Notepad, which is built into the Author program. It works much like a plain-text editor like the Windows Notepad or the DOS Editor, but uses Unicode exclusively.

The Scratchpad window differs from the other Author windows. When you save a document file in the Scratchpad, WinCALIS does not generate a *.WCA file. Without this file, WinCALIS Author does not know which Interactive (Question) Window pertains to which anticipated response. Consequently,

manually written scripts or scripts modified in the Scratchpad can be understood by WinCALIS, but cannot be interpreted by WinCALIS Author. In other words, modifications made in the Scratchpad will not be visible in WinCALIS Author.

Note: If you create a new text script in Scratchpad, WinCALIS will not create an accompanying *.WCA file.

Pass Through Window

A Tool for Advanced Authoring

The Pass Through Window provides the experienced author with the means to write statements, commands, or entire tasks, written in CALIScript, directly to the WinCALIS script. Valid CALIS commands (keywords in capital letters enclosed in angle brackets, e.g., <BLANK>, or <INSERT:gluetask2--gluetask18>) may in fact be entered by the author at any time in any CALIS window. But sometimes this is not enough, as when it is necessary to write an entire CALIS statement to the script.

Any text written in the Pass Through Window “passes through” unaltered to the *.CAL script file, when the script is generated from the *.WCA author database. It is placed by WinCALIS Author in the *.CAL file slightly indented, after the # task label statement and before the question and answer statements. No CALIS statement markers (# @ ! / ? + - = \$ & C K % or L) are inserted in the *.CAL script file preceding the lines in the Pass Through Window. Hence it allows an author to take control of the script, overriding the user-friendly, but sometimes confining automatic script generation of WinCALIS Author.

Example 1: Short Answer Responses to True/False or Multiple Choice Questions

For example, suppose you wish to add a neutral response to a True/False question, allowing the student to view some text or listen to an audio clip, in response to pressing a certain key or clicking on a button. You naturally would like to take advantage of the Author's automatic formatting of the True/False question screen, which saves you much tedious effort. But the True/False answer/feedback dialog does not permit adding a short answer neutral response like the one you need. So you can pass the needed statement directly through to WinCALIS by writing it in the Pass Through Window for the True/False task. The following CALIS statement would display an imported file to the student, if he pressed the “m” key:

```
$<KEY 'm'>;<IMPORT:tutortxt.tut>
```

Example 2: Changing Default Script Options in Mid-Script

Or, for example, you might wish to change the default script options, starting with a particular task, for the remainder of a lesson, rather than having to select the desired options separately for each remaining task. Let us say that you have turned off scoring for the first part of the lesson, in which you were presenting information to the student, but for the rest of the lesson, a review quiz to consolidate the student's new knowledge, you would like to enable scoring again.

Script options are set once by WinCALIS Author in a default option statement beginning with the “@” marker at or near the top of the script. Thus Author generated a default statement like this at the top of your script:

```
@<NOSCORE>
```

But you may reset these global options at any time during the script with a new default option statement. So in the task before the one in which scoring is to be resumed, you type the new default option statement in the Pass Through Window:

```
@<SCORE>
```

Note: Since the lines entered in the Pass Through Window are written to the script after the # task label statement for a given task, they will only take effect beginning with the next task in the task list.

The Pass Through Window is a good place to use features which are supported by WinCALIS but not automated by the Author program. WinCALIS variables, loops, and branching are all examples of such powerful advanced features.

Example 3: Declaring and Initializing CALIS Variables in the Pass Through Window

WinCALIS variables can be numeric variables, used as counters or flags, of the form

```
%index1=0
```

or they can be string variables, of the form

```
%president$="Clinton"
```

See Also: The Variable Declaration Statement

Example 4: Entering Entire Extra Tasks in the Pass Through Window

```
?These statements replace whatever was originally  
?in task 3 in Author.  
?Press ENTER to continue.  
+<ENTER>  
#3a  
?This is a complete extra task  
? which will be executed between tasks 3 and 4 and which  
?will continue in 5 seconds with the statements of the original task 3.  
?<DELAY:5><BLA>
```

You might insert in the Pass Through Window a boilerplate menu task. For an example of such a menu template, see the discussion of the <KEY controlkey> special response in the section Promoting the Status of a Particular Key (Combination).

For complete information on the syntax of CALIScript, see the section CALIScript.

A Handy Margin for Lesson Annotation

The Pass Through Window is also a good place for the author to write comments and notes during lesson development, since the contents of the Pass Through Window for each task are saved along with the *.WCA author database and included in the *.CAL script file at script generation time. Much like a memo pad for the author, the Pass Through Window will save an author's notes that can be reviewed at a later date; a student will never see the notes contained in a Pass Through Window (but see below for how to prevent that from happening).

Note that the function of the Pass Through Window is quite different from that of the Scratchpad window, the contents of which are never saved in the *.WCA author database, nor generated as part of the script created in the other windows. The functions of the Scratchpad and Pass Through Windows are thus complementary.

Because everything in the Pass Through Window is passed through directly to the script, care should be taken that no comment lines begin with capital "C," capital "K," or capital "L," which have special meaning in CALIS syntax. By preceding each comment line with a colon (:) or some other symbol which is not part of CALIScript, potential confusion can be avoided.

When used for commenting, annotating, and documenting scripts--that is, for lines which are not intended to be executed by the WinCALIS command interpreter, it is recommended practice to start every line with a symbol which has no special meaning in CALIScript, such as a colon (:), so as to avoid unexpected results which could come from WinCALIS's executing a line beginning with a capital "K" (the Keyboard Statement marker), a capital "C" (Lesson Description statement marker), a capital "L" (Default Font Statement marker) or some other CALIS statement marker.

Samples of Annotation in the Pass Through Window

Here are some sample in-script comments:

(1)

:Charlie-- We still need to add more wrong answer feedback to
:this task. --Tomoko 97.2.13

(2)

:BEGINNING OF UNIT 2
:If student answers incorrectly more than twice , use <GLUE> to
:glue in remedial tasks from script revu101.glu.
:Last revised 3/17/97

WinCALIS Author Menus

File Menu
Edit Menu
L: [Language] Menu
Format Menu
Script Menu
Task Menu
Tools Menu
Window Menu
Help Menu

The File Menu

The File item on the menu bar provides all the necessary file management options, from starting a new WinCALIS Author lesson to saving it.

Script (*.wca):

The menu items under the "Script [*.wca]" heading are relevant when you use Author to create your lesson via the standard windows (Question, Directions, etc.). As a beginner to WinCALIS, you will author lessons using Script options. It is not necessary to use functions described under the "Document file" submenu while you are first learning to author. When you create a lesson, WinCALIS Author will automatically create two files, a *.WCA Author database file and a *.CAL script file. The *.CAL file, which can be accessed under Scratchpad, contains both the lesson text and the commands necessary for WinCALIS to execute a lesson. The *.WCA file stores information about the structure of each lesson. For example, the *.WCA file keeps track of which set of answers belongs with each question. The following options appear under the Script *.WCA heading in the File pull-down menu:

[File, Script] New

When you want to begin working on a new WinCALIS Author lesson, you first have to open the File pull-down menu and select New (Alt+F,N and ENTER) under the Script *.WCA heading. This will prompt Author to begin a new file to store your commands. If you are working on another lesson at the time and you select New, WinCALIS Author will ask if you want to save the current lesson first before opening a new file.

[File, Script] Open...

The Open command allows you to select a previously saved WinCALIS Author file from a specified directory so that it may be examined or edited. When you are finished with the file, it may be saved again and reopened at any time. You must specify a path name for the lesson or scroll through directories to make the selection.

To open an existing document:

- From the File menu select the menu item Open (Alt+F,O) under Script [*.wca].
- In the File Name box, type or select the document you want to open. If the document you want to open is not listed in the File Name box, do one or more of the following:
 - a. In the Drives box, select a new drive.
 - b. In the Directories box, select a new directory.
 - c. In the List Files of Type box, search for *.WCA document.
 - d. Select OK.

See Also: File Menu, Script (*.wca): above.

[File, Script] Save F4

When you open an existing author database file for a lesson, WinCALIS Author copies it from the disk where it is stored into RAM memory and displays it. As you work, you are actually making changes to a copy of the lesson file in RAM. To ensure that your latest work is stored safely on the disk, you should frequently save your document. (This is easy to do--just press the shortcut key F4!) To permanently store the contents of your lesson, you must save it. This means that the information becomes recorded on some permanent media, such as a floppy disk, hard drive, or network storage system, and thus is retrievable for future use or modification. The lesson will be saved to the specified directory under the current name of the file. When selecting Save for the first time, you must provide certain information in order to tell WinCALIS Author where to save the file.

See Also: Document file... Save and Save As...

<p>Note: If you want to work in WinCALIS Author, be sure that you choose File Open and Save for a Script (*.WCA) file. If you modify an existing *.CAL file, WinCALIS will save the changes in its *.WCL file, but not its *.WCA file. Therefore, the modifications will not appear in WinCALIS Author, although they will alter the lesson in WinCALIS. By modifying the *.CAL file, you may lose the necessary coding information which enables WinCALIS Author to keep the lesson in its proper order. If you choose to edit a *.CAL file, we highly recommend that you save the modified version under a different name, so that you do not inadvertently write over an existing file and lose information.</p>

To save a new, unnamed document:

- From the File menu, choose Save (Alt+F,S) or (F4) OR
- From the File menu, choose Save As (Alt+F,A)
- Type a name of up to eight characters in the File Name box which pops up. WinCALIS Author will automatically add the *.WCA extension if you do not specify one.
- Click on the OK button.

To save a new/unnamed document in a different drive or directory:

- From the File menu, choose Save (Alt+F,S) or (F4) OR
- From the File menu, choose (Alt+F,A)
- In the Drives box, select a drive.
- In the Directories box, select a directory.
- Type the filename or the complete path in the File Name box.
- Select OK.

[File, Script] Save As...

If you want to rename a previously saved lesson, make a backup copy, or save it in a new place, use the Save As... option. You may then use the Windows File Manager program to delete the old version if you so choose.

To use Save As:

- From the File menu, choose Save As (Alt+F, A).
- Highlight or type the name you want to save the file under.
- Click on the OK button.

To save the script in a different drive or directory:

- From the File menu, choose Save As (Alt+F,A).
- In the Drives box, select a drive.
- In the Directories box, select a directory.
- Type the filename or the complete path in the File Name box.
- Select OK.

See Also: File Menu, Script (*.wca): above.

Document file:

As you become more comfortable with WinCALIS, you may exercise several options under the “Document file:” item in the File menu. The items under “Document file:” in the File menu allow you to create, open, or save any Unicode text file. You may also import or export files in ANSI, IBM extended ASCII (DOS), or other formats. For example, you may open a *.CAL file to view or modify a WinCALIS script. You also use the options under “Document file:” when you create ancillary files, such as tutorials, for your scripts. If you wish to create a simple lesson on paper, you may create a new file in the Scratchpad, using it as a simple word processor. Or, you may want to view a script in its entirety (including Author-generated commands) in order to better understand CALIScript process, to manually enhance a script, or to view it before printing it.

Note: If you modify an existing *.CAL file under the Document heading, WinCALIS will save the changes in its *.WCL file, but not its *.WCA file. Therefore, the modifications will not appear in WinCALIS Author, although they will alter the lesson in WinCALIS. By modifying the *.CAL file, you may lose the necessary coding

information which enables WinCALIS Author to keep the lesson in its proper order. If you choose to edit a *.CAL file, we highly recommend that you save the modified version under a different name, so that you do not inadvertently write over an existing file and lose information.

[File, Document] New

When you are already working in the Scratchpad or another window and you wish to quit working on the current document and begin working on a new Unicode text-based file, you must first open the File menu and select New (Alt+F, W) under the heading Document File, in order to tell the program that it must prepare a new file to store your commands. If you are working on another document at the time and you select New, WinCALIS Author will ask if you want to save the current document first before opening a new file.

[File, Document] Open...

The Open command allows you to pick a previously saved file from a specified directory so that it may be examined or edited. When you are finished with the file, it may be Saved and reopened at any time. You must specify a path name for the lesson or scroll through directories to make the selection.

To open an existing document:

- From the File menu select Open (Alt+F,N) option under Document File.
- In the File Name box, type or select the document you want to open. If the document you want to open is not listed in the File Name box, do one or more of the following:
 - a. In the Drives box, select a new drive.
 - b. In the Directories box, select a new directory.
 - c. In the List Files of Type box choose the type of heading which matches the format of the file. You can open the file as a Unicode/WC 2.0 file, a BOM-headed file, a No Header file, or an ANSI/WinCALIS 1.1 file, among others.
 - d. Select OK.

WinCALIS Author can import into WinCALIS format files in the following file formats:

Unicode/WC 2.0
Unicode/BOM
Unicode/No-Header
Unicode UTF-8
Unicode UTF-7
ANSI/WinCALIS 1.1
Other conversions

Unicode/WC 2.0

- For saving files in Unicode format with the WinCALIS 2.x header (including the Unicode “Byte Order Mark” or BOM), identifying it as a file to be read by the WinCALIS authoring system.

Unicode/BOM

- For saving files in Unicode format preceded by the Unicode “Byte Order Mark,” (to facilitate compatibility between machines with Intel (“little-endian”) processors and machines with Motorola (“big-endian”) processors).

Unicode/No-Header

- For saving files in Unicode format, without the WinCALIS 2.x identifying header or the BOM.

Unicode UTF-8

- For opening files in Unicode UTF-8 format.

Unicode UTF-7

- For opening files in Unicode UTF-7 format.

ANSI/WinCALIS 1.1

- For saving ANSI-standard (ISO 8859-1) single-byte plain-text files to be read into a Windows/DOS-based product (for example, Microsoft Write for Windows or any DOS Text editor).

Other conversions

- For saving files in a variety of other single-byte, double-byte, or mixed single-and-double-byte national coding standards, such as are used in localized versions of DOS and Microsoft Windows.

For more information on “Other conversions,” see the section WinCALIS Conversions.

[File, Document] Save Shift+F4

When you open an existing document, WinCALIS Author copies it from the disk where it is stored into RAM memory and displays it. As you work, you are actually making changes to a copy of the document in RAM. To ensure that your latest work is stored safely on the disk, you should frequently save your document. (This is easy to do--just press the shortcut key Shift+F4!) To permanently store the contents of your document, you must save it. This means that the information becomes recorded on some permanent media, such as a floppy disk, hard drive, or network storage system, and thus is retrievable for future use or modification. The document will be saved to the specified directory under the current name of the file. When selecting Save for the first time, you must provide certain information in order to tell WinCALIS Author where to save the file.

To save a new, unnamed document:

- From the File menu, choose Save (Alt+F,E) or (Shift+F4) OR
- From the File menu, choose Save As (Alt+F,V) or (Ctrl+F4).
- Type a name of up to eight characters in the File Name box which pops up. (WinCALIS Author will automatically add an extension if you do not specify one.)
- Click on the OK button.

To save the document in a different drive or directory:

- From the File menu, choose Save (Alt+F,E) or (Shift+F4) OR
- From the File menu, choose Save As (Alt+F,V) or (Ctrl+F4).

- In the Drives box, select a drive.
- In the Directories box, select a directory.
- Type the filename or the complete path in the File Name box.
- Select OK.

[File, Document] Save as... Ctrl+F4

To save a WinCALIS document with a different file format:

- From the file menu, under Document File select the menu item Save As (Alt+F,V or Ctrl+F4).
- In the File Name box, type a new name for the document.
- In the Save File as Type box, select the file format.
- Select OK.

WinCALIS Author can convert files in WinCALIS format to the following file formats:

Unicode/WC 2.0
Unicode/BOM
Unicode/No-Header
Unicode UTF-8
Unicode UTF-7
ANSI/WinCALIS 1.1
Other conversions

See the preceding section on the Document File, Open menu item for more information on conversion of file formats listed above. For a detailed explanation of “Other conversions,” see the section “WinCALIS Conversions.”

See Also: File Menu, Document file above.

[File, Document] Print

Selecting the File,Document File,Print menu option transfers you to one of the two WinCALIS printing utilities in order to print a *.CAL script or any document generated in the Scratchpad window or other window and saved as a text file on disk. If you wish to print directly from memory (and you are using only TrueType fonts), highlight the text to print, and choose the “Print Selection” menu option instead.

When you select “Print,” a dialog box appears which allows you to choose which utility to use: the WinCALIS TrueType Printer program (MLPTT.EXE) or the WinCALIS Multilingual Printer program (MLP.EXE). Press Enter for the default selection, the WinCALIS TrueType Printer program.

The WinCALIS Author Workstation and Student Workstation packages come with two printer programs, the WinCALIS TrueType Printer Program (MLPTT.EXE), for text written in WinCALIS Author entirely using a scaleable TrueType font, and the WinCALIS Multilingual Printer program (MLP.EXE), for text in any language and script. (Currently WinCALIS does not support TrueType fonts for Chinese, Japanese, Korean, a few other scripts such as Amharic, and miscellaneous symbol and exotic Roman character fonts.)

If your text has been written using a TrueType font (most alphabetic languages), you are advised to print with the TrueType Printer Program (MLPTT.EXE), which is faster, more powerful, and produces

superior-quality printing at any font size. If you are not sure, experiment by printing a short test document with the TrueType Printer Program. Non-TrueType characters will merely not display properly or not display at all on the printed page.

If you cannot print your print job successfully with the TrueType Printer, you should resort to printing with the Multilingual Printer. This printer program is capable of printing any text which is displayed on your WinCALIS Author screen, in any language or script, by printing all text, whether originally a raster font or a TrueType font, as graphics images.

You can also start either printer program by clicking on its icon in the WinCALIS Pangaea group.

For more information on printing, see also: Printing a WinCALIS *.CAL Script or Other Document

[File, Document] Print Selection

Selected Text (marked and highlighted in reverse video) will be printed directly from memory by the WinCALIS TrueType Multilingual Printer program.

Note: The Print Selection feature will not work properly for non-TrueType fonts on laser printers using the Hewlett-Packard PCL printer control language. This includes Chinese-Japanese-Korean, Amharic, text with IPA and other exotic Roman letters, and certain other languages supported by WinCALIS for which TrueType printing fonts are not available.

[File] Create Keyboard...

Creates a *.WCK binary language keyboard file from its ASCII text counterpart *.MAP file opened in the Scratchpad as an ASCII/ANSI file.

See Also: Defining Language Keyboard Files

[File] Create Global File...

Creates a new WINCALIS.WGI binary global information file from its ASCII text counterpart WINCALIS.INF, opened in the Scratchpad as an ASCII/ANSI file.

See Also: The WinCALIS Global Information File WINCALIS.WGI

[File] Exit

When you are finished working with WinCALIS Author, you should select the Exit command. If you have made any changes to the current lesson, WinCALIS Author will ask if you want to save the file.

The Edit Menu

[Edit] Undo Alt+BkSp

Undo allows you to delete the last editing change made to the text and recall any affected text. For example, if you delete a paragraph from your text and then change your mind, you can employ the undo feature to bring back the paragraph. Undo may be used to undo an entire series of editing changes.

[Edit] Redo Alt+Shift+BkSp

Undoes the effect of Undo and restores the previous editing change, or series of editing changes.

[Edit] Cut Ctrl+X

To cut a highlighted text block means to delete the block from the lesson and copy it to the clipboard, a temporary place where text is stored for later reinsertion using the Paste command.

The shortcut keys for Cut are Ctrl+X or Shift+Delete.

[Edit] Copy Ctrl+C

Copying a highlighted text block is like cutting it, except the block remains in place in the activated window. A copy of the selected text is placed on the clipboard so that it may be replaced in another location or window.

The shortcut keys for Copy are Ctrl+C or Ctrl+Insert.

[Edit] Paste Ctrl+V

Paste takes whatever information is on the clipboard put there by the Cut or Copy commands (either WinCALIS Author Unicode text or another application's plain ANSI text) and inserts the text at the insertion point (where the blinking “caret” is located).

The shortcut keys for Paste are Ctrl+V or Shift+Insert.

[Edit] Paste as...

Use Paste as... to paste non-Unicode text on the clipboard which was put there by another application into WinCALIS Author at the insertion point (where the blinking “caret” is located), converting the codes from the other application's format to Unicode.

When you choose Paste As..., a “Convert From” dialog box appears, permitting you to select the code to convert from.

For possible code conversions, see
WinCALIS Conversions

[Edit] Delete Del

Delete removes the highlighted block from the active window, but does not place it on the clipboard. Text may immediately be restored using the Undo command.

[Edit] Select All

Select All highlights all the text in the active window so it may be cut, copied or deleted. You may highlight text easily by placing the arrow at the starting point, clicking and dragging the highlight to the end point of the text block you wish to select.

[Edit] Find...

You can search for each occurrence of a combination of any characters, including uppercase and lowercase characters, whole words, or parts of words using the Find option.

To find text:

- From the Edit menu, choose Find (Alt+E,F).
- In the Find box type the text, word(s), or combination of characters you want to find.
- Set the options you want to control the search.
- Choose the Find Next button. If an occurrence is found, the screen scrolls so you can see the text in context.

Match Whole Word Only

Finds occurrences of the text as a whole word (no partial words). For example, the input “Tube” would not match up with the text “Tubes” if the “Match Whole Word only” option had been selected.

Match Case

Finds text having a certain pattern of uppercase and lowercase letters.

Note: A case sensitive search is much faster than a non-case sensitive search, since in the latter case, Author must consult the WinCALIS Global Information File for information on upper- or lower-case equivalents for every character in the text--even for those scripts which have no case distinction. Thus Match Case is checked by default.

Up

Searches the document above the insertion point.

Down

Searches the document below the insertion point.

Cancel

To cancel a search or when you are finished searching.

Note: To search for text in a non-ANSI character set, first select the appropriate keyboard from the L: [Language] menu.

Find Next F12

You can search for the search string most recently searched for with the Find option by pressing the shortcut key F12 or by selecting the Find Next menu option.

The shortcut key for Find Next is F12.

[Edit] Replace...

With the Replace command, you can find and replace text. You can automatically replace all occurrences of the found text, or selectively replace only certain instances. For example, if you wanted to replace the word “beast” with the word “monster” you would simply type “beast” in the Find What box, go to the Replace With box using the mouse or Tab key and enter the word “monster.” If you wanted to replace every occurrence of the word, you would select Replace All.

To replace text:

- Select the portion of the document you want to search. WinCALIS Author editor searches the entire document if you do not select a portion of the document.
- From the Edit menu, choose Replace (Alt+E,R).
- In the Replace box, type the text to be replaced.
- Set the options you want to control the search.
- To replace text, choose Replace.
- To leave text unchanged and search for next occurrence, choose Find Next.

- To cancel a Replace in progress or when you have finished replacing, select Cancel.

Match Whole Word Only

Finds occurrences of the text as a whole word (no partial words)

Match Case

Finds text having a certain pattern of uppercase and lowercase letters (a case-sensitive search).

Note: A case sensitive search is much faster than a non-case sensitive search, since in the latter case, Author must consult the WinCALIS Global Information File for information on upper- or lower-case equivalents for every character in the text--even for those scripts which have no case distinction. Thus Match Case is checked by default.

To replace the search text with nothing, make sure the Replace With box is empty by deleting the contents of the box.

To undo the effects of the Replace command, choose Undo from the Edit menu as the first action after completing the change.

[Edit] Add Accent F2

Selecting the menu item Add Accent or pressing the F2 function key highlights the character to the left of the insertion point (blinking caret) and prepares to overstrike that character with an accent determined by the next key pressed. See the section Accents and Diacritics.

[Edit] Go to Line...

Moves the cursor to the line number (Alt+E,L) you specify in the Go to Line # edit box.

[Edit] Switch Languages F3

Switches between the currently selected language and the “top” language keyboard for the workstation (by default English/W.Europe) by using the function key F3.

Note: The “top” language is determined by the default language .INI file defined in the “DefaultIni=” line in the [WinCALIS] section of the main WinCALIS initialization file “WinCALIS.ini,” in the WinCALIS program directory. The default language .INI file is set by default for the United States and Western Europe to “latin1.ini.” If the default .INI file is changed to some other language .INI file, the “top” language will also be changed accordingly. The initial language for use on startup in WinCALIS and WinCALIS Author (also by default set to English/W.Europe) is also customizable by the courseware author or WinCALIS system administrator. The initial language is determined by the setting of the InitialLanguage= line in the [Initial] section of the “WinCALIS.ini” file. The out-of-the-box value for this line is Initial Language=English/W.Europe.

[Edit] Unicode Intput... (Ctrl+I)

Allows you to browse through the Unicode character tables and insert a character in your text (Alt+E,I or Ctrl+I).

See Unicode Character Selection Dialog Box.

[Edit] Insert

The Insert submenu allows the author to place non-displaying control characters, such as joiners, zero-width spaces, and other special characters in text, to achieve special effects such as override language directionality.

Joiner Shift+F5

This character can be used to join with other characters so as to maintain certain contextual forms that would not otherwise occur during ordinary editing. For instance, if a character such as an Arabic letter has an initial, medial, and final form, you may insert joiners in order to force the system to maintain the medial or final forms out of context.

The shortcut key to insert a Joiner is Shift+F5.

Non-Joiner Shift+F6

The Non-Joiner works in much the same way as the joiner. You may use this option to prevent two characters from joining when they otherwise would.

The shortcut key to insert a Non-Joiner is Shift+F6.

Left to Right Override Shift+F7

Right to Left Override Shift+F8

These option allows you to manually determine the direction in which text will appear on screen. This could be useful if, for instance, you were writing an English sentence which started with a Hebrew word, and you wished to force the sentence to be left-justified, or conversely, a Hebrew sentence which began with an English word, which you wished to be right-justified.

The shortcut key to insert a Left-to-Right Override is Shift+F7.

The shortcut key to insert a Right-to-Left Override is Shift+F8.

Zero Width Space Shift+F9

This character is a no-width space character which can be used to identify word boundaries in languages that do not have spaces between all words.

The shortcut key to insert a zero-width space is Shift+F9.

[Edit] Word Wrap

Allows you to wrap text (Alt+E,W); text retrieved from disk or in any line being edited is wrapped at the right margin of the maximum window size for the screen resolution of your system, as reported by Windows.

The L: [Language] Menu

General Language and Keyboard Information

The language pull-down menu (Alt+L) allows you to select any of the language keyboards available in WinCALIS Author. Depending on the number of fonts and keyboard layouts available on your system, the number of languages may vary from just a few to most of the languages of the world.

Western European languages such as Danish, English, French, German, Italian, Portuguese, Spanish, and Swedish are all available with the ANSI (Latin1) Roman character set and default "English/W.Europe" keyboard (no keyboard change required). All accents are typed with three keystrokes (base letter + F2 + accent).



Figure 5: Keyboard Help image

Other languages, such as Russian or Ukrainian, with the Cyrillic character set, or Arabic or Persian (Farsi), with the Arabic character set, require entirely different fonts and keyboards. A virtual keyboard can be displayed on screen by clicking on Keyboard Help in the Help menu. For an explanation of conventions used in the virtual keyboard, see Accents and Diacritics.

You may switch between the currently selected language keyboard and the “top” language keyboard for the workstation (by default English/W.Europe) by using the function key F3 shortcut key.

Under the “Europe (other)” submenu there are also language-specific keyboards for Dutch, French, German, Italian, Portuguese, and Spanish. These keyboards emulate some of the features of national typewriter keyboards (but without “deadkeys”). On these keyboards, certain symbols on the shifted number keys are replaced with accented letters, so that only one keystroke, rather than three (letter + F2 + accent), as in the default “English/W.Europe” keyboard, is required to type “ñ,” “ç,” “ß,” “ü,” “â,” etc. These keyboards are often popular with native-speaker teachers. But note that certain WinCALIS wild cards and other CALIS language symbols are sometimes sacrificed in the process. If you use one of the “Europe (other)” national keyboards, you may need to press F3 to switch momentarily to “English/W.Europe,” in order to type one of the symbols !, @, #, \$, %, ^, &, *, or |.

When you change languages in the middle of a lesson, only the new text that you type will appear in the selected language. Previously written text will not be affected.

See Also: Specific Language Reference Manuals
Switch Languages
Right-to-Left and Left-to-Right Languages

[Language] Add Language...

Adds a language keyboard item to your L:[Language] menu in WinCALIS Author and Language: menu in WinCALIS from among those available in your WinCALIS program directory (i.e., for which there exist *.INI, *.MAP, and *.WCK files).

See Also: Defining Language Keyboard Files

[Language] Delete Language...

Deletes a language keyboard item from your L:[language] menu. Only the menu listing is deleted. The *.INI, *.MAP, and *.WCK files for each language keyboard are not deleted from your WinCALIS program directory.

Note: When you install WinCALIS, it installs *.INI, *.MAP, and *.WCK files for all available languages into your program directory by default. Add Language allows you to access any available language. If you need to

economize on disk space, after deleting a language from your menu, you may also use the File Manager to delete its *.INI, *.MAP, and *.WCK files from your hard disk, or move them to a backup storage location. *.MAP files are simply the text versions of binary keyboard *.WCK files, which are those used by WinCALIS. If you do not anticipate the need to edit or add any keyboard file, you may delete *all* the *.MAP files from your system. See [Deleting Nonessential Files to Save on Disk Space](#).

The Format Menu

Characters are letters, punctuation marks, numbers, and symbols that you type as text. Character formatting determines how characters appear on the screen. You may add emphasis to text by applying such attributes as **color**, **bold**, *italic*, underline, ~~strikeout~~, and **reverse video** (foreground and background colors reversed).

[Format] Center...

Places the <CE> command (the abbreviation for <CENTER>) in your script at the cursor point, which tells WinCALIS to center everything on that line, following the command.

[Format] Tab Stop...

Places a <TAB:n> command in your script at the cursor point, which tells WinCALIS to tab over *n* number of times. Replace the *n* with the desired number of tab stops.

[Format] Blank Window...

Places a <BLA> command (the abbreviation for <BLANK>) in your script at the cursor point, which tells WinCALIS to blank the current window.

[Format] Font...

Allows you to set the font for a window.

There is a default font for every character set used by WinCALIS, as specified in the configuration file WCFNT.INF. See [The Default Character-to-Font Mapping File WCFNT.INF](#). If you prefer a different font for text in WinCALIS, you can change the font that WinCALIS uses for a particular window. You can use this menu option to insert a <SET FONT facename> command in that window at the place the font is to take effect. WinCALIS will use the new font in that window from then on, or until you choose a different font.

To change the default font:

- From the Format menu, choose Font.
- Type or select any of the fonts available to you in the Font dialog box.

Note: The selection of facenames displayed is read from the *.INI file for the currently selected keyboard. See [Rearranging the Language Menus](#). Thus if “English/W.Europe” is the current keyboard, the font names are read from the corresponding “latin1.ini” file. The fonts listed are those which the courseware author can be confident will be available on the WinCALIS Student Workstation, when the exercise is worked: Arial, Courier New, Times New Roman, MS Sans Serif, and Fixedsys--all fonts which are part of the basic Windows 3.1 system. Similarly, font selections for other language keyboards are based on those fonts actually distributed with WinCALIS.

See Also: [The “L” Default Font Statement](#)

[Format] Fontsize

The Fontsize menu allows you to set the font size for display of a single line of text.

Note: WinCALIS Author always displays text in “Medium” font size. The actual point size of the “Medium” used by WinCALIS Author is determined by the pixel height settings in the default language .INI file defined in “DefaultIni=” line in the [WinCALIS] section of the main WinCALIS initialization file “WinCALIS.ini,” in the

WinCALIS program directory. The default language .INI file is set for the United States and Western Europe to "latin1.ini." Changing the line "Medium=21" in the [FontSize] section of "latin1.ini" to some greater or smaller pixel height than 21 pixels will adjust the size of the default font accordingly in WinCALIS Author and the Author Workstation's copy of WinCALIS. All pixel heights are set by default for standard VGA resolution (640x480 pixels) as follows:

Tiny=12
Fine=16
Small=19
Medium=21
Large=32
Xlarge=48
Huge=66.

Special versions of "latin1.ini" for use with EGA monitors and with VGA monitors at higher than standard VGA resolution are distributed as part of the WinCALIS Author and Student Workstation packages.

[Format] Selection:

There are several options displayed in the Format menu under the heading Selection. These options allow you to apply particular formatting to selected text. They include color, bold, italic, underline, strikeout, and reverse video. The appropriate commands are added automatically surrounding your selected text in Author. The correct formatting will show up in WinCALIS.

[Format, Selection] Insert Marks

You may highlight a string of text or a word and then choose this option to automatically place a marked pair around it, increasing the numbers sequentially for each subsequent pair, ie. <M:m1>highlighted text<M:n1>. Here, M stands for the CALIScript command MARK. This will prepare the text to be made into "hot text" later in the script.

See: <MARK>ing for Hot Text

[Format, Selection] Color

You may add color to specific type to add emphasis. The colors available are listed in the color palette submenu, accessed by clicking on Color listed under Selection in the Format menu.

To apply color to text:

- Select text by highlighting it with the mouse.
- Click on the color name in the color palette submenu.

To apply other formatting attributes to text:

- Select text by highlighting it with the mouse.
- Select an option from those listed under the Format main heading.

Other available attributes:

[Format, Selection] Bold

Adds the necessary commands surrounding your highlighted text to **bold** the text in WinCALIS.

[Format, Selection] Italic

Adds the necessary commands surrounding your highlighted text to *italicize* the text in WinCALIS.

[Format, Selection] Underline

Adds the necessary commands surrounding your highlighted text to underline the text in WinCALIS.

[Format, Selection] Strikeout

Adds the necessary commands surrounding your highlighted text to ~~strike-out~~ the text in WinCALIS.

[Format, Selection] Reverse Video

Adds the necessary commands surrounding your highlighted text to make the text appear in **reverse video** (foreground and background colors reversed) in WinCALIS.

[Format] Set:

The options listed under Set in the Format menu allow you to change the formatting of all subsequent text in a window in WinCALIS, by inserting the appropriate commands in your WinCALIS Author text at the insertion point.

[Format, Set] Foreground Color

Changes the color of the text in WinCALIS. Set Foreground Color allows you to choose from a list of colors in a submenu accessed under the Foreground Color menu option.

[Format, Set] Background Color

Changes the window background color in WinCALIS. Set Background Color allows you to choose from a list of colors in a submenu accessed under the Background Color menu option.

[Format, Set] Fontsize

Sets the font size of the text in a window until it is changed. Set Fontsize allows you to choose from a list of font sizes in a submenu accessed under the Fontsize menu option.

Note: The actual point sizes are determined by the pixel height settings in the default language .INI file defined in "DefaultIni=" line in the [WinCALIS] section of the main WinCALIS initialization file "WinCALIS.ini," in the WinCALIS program directory. The default language .INI file is set for the United States and Western Europe to "latin1.ini." The pixel heights in "latin1.ini" are set for standard VGA resolution (640x480 pixels) as follows:

Tiny=12
Fine=16
Small=19
Medium=21
Large=32
Xlarge=48
Huge=66

They may be edited by the end user or system administrator. See Changing the Default Font Size for Use in WinCALIS Author.

[Format, Set] Font Atttributes

Sets the attribute (bold, italic, normal, strikeout, or underlined) of a font in a window, until it is changed. Set Font Attributes allows you to choose from a list of font attributes in a submenu accessed from the Font Attributes menu option.

The Script Menu

[Script] Script Options...

You control many options which contribute to the appearance and operation of your lesson. All WinCALIS options have default values to make authoring as easy as possible for you. But by

selectively changing various default options, you determine how information is conveyed to the user. The section Script (Default) and Task Options details various ways you can alter your lesson presentation using the Script Options menu.

[Script] Cloze Options...

Specifies characteristics of cloze exercises. See Cloze Options.

[Script] Script Generation Options...

Specifies characteristics of your lesson which WinCALIS Author can automatically produce for you at the time of script generation by inserting various statements and commands, rather than through the use of WinCALIS script default and task options. See Script Generation Options.

[Script] Script Start-up Directions...

Opens a special Popup (Directions) Window in which you may write directions to the student which will be executed only once, at the beginning of the script. Directions written in this window are placed by WinCALIS Author in the #0 task, when generating the *.CAL script file. If there are no other directions written in the Popup (Directions) Window for Task 1, directions placed here will appear no different to the student from those placed in the Popup (Directions) Window for Task 1. However, if directions are placed both here and in the Popup (Directions) Window for Task 1, the directions for Task 1 will appear immediately after the Start-up Directions, replacing the start-up directions in the Popup (Directions) Window.

One potential application for start-up directions placed here would be for a copyright message for your exercises.

[Script] Run Script Ctrl+R

Launches WinCALIS, or switches to WinCALIS, if it is already running, and executes the *.CAL script generated from the *.WCA author database, as it was most recently saved to disk. WinCALIS should be in Text Conversion (*.CAL) mode and the current directory the files of which are displayed in WinCALIS's Lesson Window should be the one which contains the script.

The Task Menu

[Task] Short Answer... Ctrl+S

If the Interactive (Question) Window is active, clicking on this menu option or pressing Ctrl+S selects the Short Answer question type and opens the Short Answer Answers/Feedback dialog box.

[Task] Multiple Choice... Ctrl+M

If the Interactive (Question) Window is active, clicking on this menu option or pressing Ctrl+M selects the Multiple Choice question type and opens the Multiple Choice Answers/Feedback dialog box.

[Task] True/False, True... Ctrl+T

If the Interactive (Question) Window is active, clicking on this menu option or pressing Ctrl+T selects the True/False question type, in which "True" is the Correct/Continue response and opens the True/False, True Answers/Feedback dialog box.

[Task] True/False, False... Ctrl+F

If the Interactive (Question) Window is active, clicking on this menu option or pressing Ctrl+F selects the True/False question type, in which “False” is the Correct/Continue response and opens the True/False, False Answers/Feedback dialog box.

[Task] Cloze Exercise... Ctrl+Z

If the Interactive (Question) Window is active, clicking on this menu option or pressing Ctrl+Z selects the Short Answer question type and opens the Cloze Exercise dialog box.

[Task] Next Ctrl+N

Moves you to the next task (question) in the script.

To move to the next task:

- From the Task menu, choose Next (Alt+T,N or Ctrl+N).

See Also: Moving Among Tasks with the Task Menu

[Task] Previous Ctrl+P

Moves you to the previous task (question) in the script.

To move to the previous task:

- From the Task menu, choose Previous (Alt+T,P or Ctrl+P).

[Task] Delete Ctrl+D

Removes the current task (question) from the script.

To remove a task from the script:

- From the Task menu, choose Delete (Alt+T,D or Ctrl+D).

[Task] Go to Task... Ctrl+G

Moves to any task (question) in the script.

To move directly to a task:

- From the Task menu, choose Go to Task (Alt+T,G or Ctrl+G).

[Task] Rearrange Tasks... Ctrl+A

If you have several tasks in your script, you can use Rearrange Tasks to arrange the tasks in a different order.

To rearrange tasks in your script:

- From the Task menu, choose Rearrange Tasks (Alt+T,A or Ctrl+A). The Rearrange Tasks dialog box appears.

a. To Move a Task:

- Click on the text box to the right of "Move Task Number:".
- Type the number of the task you want to move.
- Click on or tab to the "to Task Number:" text box.
- Type the number of the location to move the task to.

b. To Copy a Task:

- Click on or tab to the "Copy Task Number:" text box.
- Type the number of the task to be copied.
- Go to the "as Task Number:" text box.
- Type the number of the location the task is to be copied to.

c. To Switch Tasks:

- Click on or tab to the "Switch Task Number:" text box.
- Type the number of one of the tasks to be switched.
- Go to the "and Task Number:" text box.
- Type the number of the other task to be switched.

Once you have finished your task rearranging, choose the OK button, or to cancel the modifications, choose the Cancel button.

[Task] Change Task Type... Ctrl+K

You may use the Change Task Type option to change the type of an already defined question from or to Short Answer, Multiple Choice, True/False with a True Correct/Continue response, or True/False with a False Correct/Continue response. Cloze exercises, once selected cannot be changed into another type.

[Task] Tutorial Files...

Links external tutorial files with the F[1-10] tutorial windows.

See Also: Extra-CALIS Communication
Moving Among Tasks with the Task Menu

The Tools Menu

By embedding multimedia objects in your WinCALIS scripts, you can enhance the interactive nature of your lesson. WinCALIS Author allows you to place multimedia objects automatically at various places in your lesson.

To insert an object into WinCALIS:

- Position the insertion point where you want to embed the object.

- From the main menu bar choose Tools (Alt+O).
- Under Tools select the menu item that describes the resource to be embedded: Audio Clip, Video Clip, or Graphic.

[Tools] Audio Clip...

Inserts an audio clip in your lesson in the active window at the current insertion point.

To insert an audio clip:

- From the Tools menu, choose the item Audio Clip. An Audio Clip dialog box will appear, containing various clip names in the "Name:" list box.
- Choose the appropriate clip by clicking on a clip name in the "Name:" list box.
- Select any options you desire.
- Choose OK.

To delete an existing clip:

- Replace the clip with another clip name OR
- Highlight the clip name to be deleted and press the Delete key.

Note: Clip names which appear in the pop-up "Name:" list box must be created first using the Multimedia Editor. For more information on how to create these applications, see the section Multimedia in WinCALIS.

[Tools] Video Clip...

Inserts a video clip in your lesson in the active window at the current insertion point.

To insert a video or animation clip:

- Select Video Clip from the Tools menu. You will be presented with a dialog box containing various clip names in the "Name:" list box.
- Choose the appropriate clip by clicking on a clip name in the "Name:" list box.
- Choose the location for the clip to appear and other options.
- Select OK.

To delete an existing clip:

- Replace the clip with another clip name OR
- Highlight the clip name to be deleted in the "Name:" list box and press the Delete key.

Note: Clip names which appear in the pop-up "Name:" list box must be created first using the Multimedia Editor. For more information on how to create these applications, see the section Multimedia in WinCALIS.

[Tools] Insert Graphic...

Inserts a reference to a bitmap graphic *.BMP file in your lesson in the active window at the current insertion point.

To insert a graphic:

- From the Tools menu choose the menu item Graphic. You will be presented with a Graphic dialog box, containing various graphic names in a list box.
- Choose the appropriate graphic by clicking on a clip name in the list box.
- Choose OK.

Note: the items displayed in the Graphic dialog box are all of the *.bmp files found in:
a) the current directory (where the script is saved to) and, b) the \\wincalis\\graphics subdirectory.

To delete an existing graphic:

- Replace the current graphic with another graphic name OR
- Highlight the selected graphic in the list box and press the Delete key.

[Tools] Insert Button... Ctrl+B

Displays the Create Button dialog box, enabling you to insert one or buttons in the active window of the current task in your script (except Popup Directions Window). Button location on screen, text on button, button features, button action when pressed, and cursor position after button is created may all be edited in the Button dialog box. Special responses for processing button input are automatically added to the list of Anticipated Responses you create using the Task menu.

As with audio and video clips inserted with the Audio Clip and Video Clip dialog boxes, neither the buttons themselves nor the commands to make them appear and give them the desired attributes in WinCALIS are shown in the WinCALIS Author window. The commands generated, however, may be viewed and fine-tuned by opening the .CAL file in the Author Scratchpad.

See Also: Creating Buttons and Hotspots

[Tools] Insert CALIScript Command...

This option allows you to write any CALIScript command, special response, or wild card character to your script, by choosing the generic item from the Insert CALIScript Command list, and modifying it if needed to suit your purpose. For those items that can be modified/customized, variables are given as models. For example, after inserting the command

<GRAPHIC:X.bmp [name]>

then you can specify the exact *.bmp graphic file in place of the X, and can choose to give the graphic a labelname if you plan to enable the graphic hotspot feature later in the script. For example:

<GRAPHIC: georgia.bmp hotlanta>

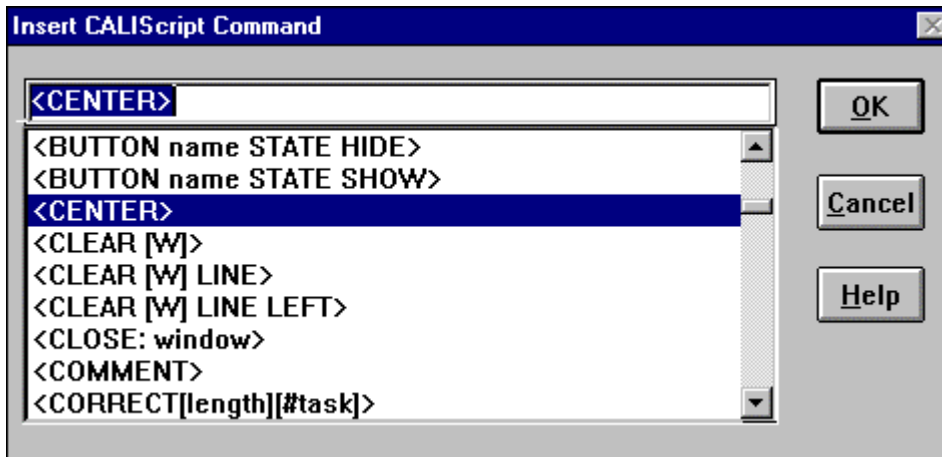


Figure 6: Insert CALIScript Command Tool

[Tools] Create Hot Text

Displays the Create Hot Text dialog box, enabling you to mark strings of text in your script as “hot” semi-automatically, by inserting CALIS `<MARK:x>` commands around the text and linking each instance of hot text to a `<MOUSE: x y>` Special Response and an action in response to that mouse click which you specify. When the user clicks the mouse on the hot text, WinCALIS will carry out the action specified, such as playing an audio clip, popping up some explanatory text in a feedback window, or making a hypertext jump to a new task.

Special Responses are automatically added to the list of Anticipated Responses you create using the Task menu. The Special Responses generated do not appear in the Anticipated Responses/Feedback dialog box, but may be viewed and fine-tuned by opening the .CAL file in the Author Scratchpad.

See Also: Using Special Responses: `<MOUSE: xy>`
Positioning the Cursor at Specific Marks in Your Text or Question

[Tools] Multimedia Editor...

Launches the separate WinCALIS Author Workstation utility “Multimedia Editor,” for the purpose of viewing and marking clips in media files and simple devices. You may also start the Multimedia Editor by clicking on its icon in the WinCALIS Pangaea Group in Program Manager.

The Multimedia Editor provides a flexible, independent interface for creating the various multimedia objects that you can incorporate into a lesson. It enables you to control and integrate analog and digital video, audio, and animation clips.

To run the Multimedia Editor:

- From the main menu choose Tools (Alt+O,M).
- Choose Multimedia Editor...

Detailed information on the Multimedia Editor and embedding various multimedia modules into a lesson is found in the section Multimedia in WinCALIS..

Note: In order for the Clips you defined to be displayed, you must first save your script with the same name as the *.MMD file using the Script [*wca], Save option. For example, if you saved your clips under the heading ELVIS.MMD, you would save your script as ELVIS.WCA in WinCALIS Author.

[Tools] Language Options

Editing options which are specific to particular languages are grouped under this menu item. Each language has its own cascading submenu.

[Tools, Language Options] Chinese Options

Reselect ("O <u>o</u> ps")	F9
<u>C</u> haracter Search	F7
<u>A</u> ll-tone Search	F8
Default Selection/ <u>R</u> epeat	F10
Set <u>C</u> ache/No-Cache Search Mode	Alt+F3
Set <u>C</u> onvert/No-Convert Mode	Alt+F5
Set <u>W</u> ord/Character Search Mode	Alt+F7
Set <u>T</u> one/All-Tone Search Mode	Alt+F8
Set <u>S</u> imple/Complex Style	Alt+F9
Set <u>D</u> efault/No-Default Search Mode	Alt+F10
Set <u>W</u> ord/Character Search Mode	Alt+F7

These menu options alter the Chinese pinyin-Hanzi conversion process. For an explanation of the various Chinese conversion settings and search options, see the topic Chinese Function Keys or the general Chinese Language Reference (requires separate CJKV module).

More Options... Alt+F11

If the current language keyboard is Chinese Pinyin-to-Hanzi, selecting the menu item Tools, Language Options, Chinese Options, More Options will display the Chinese Dictionary Search Options dialog box, with the current settings for Chinese input. These settings can be changed either by clicking with the mouse or by pressing the corresponding Alt + function key combination.

Set Disk/No-Disk Search Mode Alt+F12

As with the other menu options above this option alters the Chinese pinyin-Hanzi conversion process. See the topic Chinese Function Keys or the general Chinese Language Reference (requires separate CJKV module).

Chinese Status Bar

When you choose Chinese Pinyin-to Hanzi as the active language keyboard, the Chinese Status Bar will pop up by default, displaying the current settings for Chinese input.

The Status Bar may be turned off by deselecting it here. If Chinese Pinyin-to-Hanzi is not the currently selected language, the menu item is dimmed.

[Tools, Language Options] Japanese Options

View Japanese Dictionary...

Allows you to view and edit the Japanese internal Kana-Kanji conversion dictionary. (Currently disabled.) See Japanese (requires separate CJKV module).

[Tools, Language Options] Korean Options

Hanja-to-Hangul* *F7

To convert the syllable or word to the left of the insertion point (the blinking caret) written in Hangul into its Hanja equivalent, press the F8 function key, or click on the menu item Tools, Language Options, Korean Options, Hangul-to-Hanja.

Hangul-to-Hanja* *F8

To convert the syllable or word to the left of the insertion point (the blinking caret) written in Hanja into its Hangul equivalent, press the F7 function key, or click on the menu item Tools, Language Options, Korean Options, Hanja-to-Hangul.

See Korean (requires separate CJKV module).

[Tools] Look Up Word... *F5*

If one or more on-line dictionaries have been supplied to accompany your copy of WinCALIS Author, selecting this menu option will open a dialog box, allowing you to look up the meaning or pronunciation of a selected word or phrase. Depending on how the dictionary was installed, it is either accessible only from specific language keyboards (e.g. for a Korean-English dictionary, the current keyboard must be one of the Korean keyboards), or accessible from any language keyboard which does not have a specific dictionary linked to it (e.g. for a Korean-English dictionary, the current keyboard could be either the default English/W. Europe keyboard or any of the Korean keyboards).

In order to select a word or phrase to look up, drag the mouse across it (or double click on it if it is a single word) to highlight it. Then press F5 or click on the menu item Tools, Look Up Word... to look up the word. Repeat as often as desired.

You can press the Escape key at any time during a search to cancel further searching and display the matches found thus far.

See Also: User-Supplied On-line Dictionaries

[Tools] Sort Lines...

Sorts the contents of the active window by line in Unicode (alphabetical) order. Before you sort the lines in a window, be sure that you save a copy of your file on disk, to avoid unintended results. Once a window is sorted, it cannot be unsorted, except by hand.

[Tools] Generate Index...

This menu option in WinCALIS Author or UniEdit runs the Unicode index builder to build an index for a dictionary file in the default "Dictionaries" subdirectory. The dictionary should be saved as type "Unicode/No Header".

See How To Create Your Own Dictionary

[Tools] Set Tools...

This menu option displays the Set Tools dialog box, enabling you to add, delete, or move menu items on the Tools menu. From the WinCALIS Student Workstation it is also possible to add, modify, and delete items in your Tools menu by editing the [Tools] section of your WINCALIS.INI file with a plain-text editor, such as the Windows Notepad or DOS Edit program. Be sure to make a backup of your original WINCALIS.INI first.

[Tools] Check Spelling

If a spelling checker has been installed on your workstation and linked to WinCALIS Author, selecting this option will start the spelling checker. A copy of the spelling checker WinSpell® 3.0 is included with the WinCALIS Author workstation setup.

[Tools] Print File (TrueType)

Selecting this item launches the WinCALIS Author TrueType Multilingual Printer Program.

[Tools] Print File (Bitmap)

Selecting this item launches the WinCALIS Author Graphics Multilingual Printer Program.

[Tools] Windows Paintbrush

Selecting this menu option launches the Windows Paintbrush (Windows 3.1) or Windows Paint (Windows 95) graphics tool with the default WinCALIS graphics directory as the working directory, making it easier to view and edit the graphics files therein.

[Tools] Keyboard Macro

Selecting this item launches the Windows Macro Recorder accessory, enabling you to define a keyboard macro to type automatically and rapidly a complex or repetitive series of keystrokes. A predefined dummy macro is loaded automatically to help you get started, if you wish to define a macro to playback a repetitive series of keystrokes.

For example, if you need to add a particular WinCALIS command or commands (e.g., <DRAW box1 BOX 100,200,1,BLUE YELLOW> or <FS FINE><BO>) in a variety of places at periodic intervals, so that it is inconvenient to copy it to the clipboard and paste it. Or if you need to perform a complex operation such as finding certain text and deleting certain other adjacent text (the undocumented F12 “Repeat Find” key in Author is also handy in such cases). Or type a Special Response in the Anticipated Answer box of the Answers/Feedback dialog, then Tab down and type the desired reaction to it in the Feedback box.

Note that for beginners, it is easier to choose to ignore the mouse and just record keystrokes. But for experienced users, you can make very complex macros using just the keyboard or the keyboard and the mouse. For more information, see the On-line help for the Recorder, or your Windows manual.

[Tools] Tools Menu Readme

Selecting this item displays information about the Tools menu and some of its functions.

The Window Menu

[Window] Tile Horizontally

Arranges open windows above and below each other in the WinCALIS Author editor screen.

[Window] Tile Vertically

Arranges open windows side by side in the WinCALIS Author editor screen.

[Window] Cascade

Arranges open windows partially overlapping, with their title bars visible.

[Window] Arrange Icons

Arranges the icons for minimized windows across the bottom of the screen

[Window] Desktop

WinCALIS Author provides you with a customizable user environment. The Desktop menu options allow you to customize your Author desktop, including such aspects as which windows are open, their size and shape, whether or not to display the status bar at the bottom of the screen. the default startup language keyboard, and Chinese editing and dictionary search options.

[Window, Desktop] Save

Allows you to save the current desktop.

[Window, Desktop] Restore Ctrl+7

Restores the Initial desktop.

[Window, Desktop] Default Ctrl+8

Restores the default desktop configuration as installed by the Author Workstation Setup.

Note: WinCALIS Author saves the desktop you have arranged while editing a lesson as part of the *.WCA file for that lesson.

[Window, Desktop] Status Bar

Selecting this item displays a Status Bar at the bottom of the screen.

[Window] 1 Interactive (Question) (Ctrl+1)

Clicking on this menu option or pressing the shortcut key Ctrl+1 switches to the Interactive (Question) Window.

[Window] 2 Presentation (Text) (Ctrl+2)

Clicking on this menu option or pressing the shortcut key Ctrl+2 switches to the Presentation (Text) Window.

[Window] 3 Popup (Directions) (Ctrl+3)

Clicking on this menu option or pressing the shortcut key Ctrl+3 switches to the Popup (Directions) Window.

[Window] 4 Lesson Description (Ctrl+4)

Clicking on this menu option or pressing the shortcut key Ctrl+4 switches to the Lesson Description Window.

[Window] 5 Pass Through (Ctrl+5)

Clicking on this menu option or pressing the shortcut key Ctrl+5 switches to the Pass Through Window.

[Window] 6 Scratchpad (Ctrl+6)

Clicking on this menu option or pressing the shortcut key Ctrl+6 switches to the Scratchpad Window.

Unsaved Asterisk *

When changes are made to any window, an asterisk * will be added to the title bar to indicate that changes have not yet been saved. Once the document or script is saved, then the asterisk * will disappear. Remember that the Scratchpad window acts independently of the other script windows.

Arranging Windows and Icons

WinCALIS Author enables you to arrange your desktop so that windows and icons are easy to see. The Tile Horizontally option under the Window menu arranges open windows above and below each other in the WinCALIS Author editor window. The Tile Vertically option arranges the windows side by side in the WinCALIS Author editor window. The Cascade command resizes and layers windows so that each title bar is visible.

To arrange windows on your desktop:

- Choose Cascade or Tile from the Window menu.
- Use the Arrange Icons command to evenly arrange the icons in a group window.

To arrange program-item icons:

- Choose Arrange Icons from the Window menu.

To activate the editing window in WinCALIS Author:

- Using the mouse, click anywhere in one of the WinCALIS Author windows *or*
- Select Window from the menu bar, and click on one of the six available editing windows (Ctrl + 1-6)

The Help Menu

In addition to the menu options explained below, WinCALIS Author has the following additional context-sensitive Help features, available from the keyboard:

Keyword Search (Shift+F1)

Put your caret anywhere on a CALIS command, task option, special response, or global variable keyword (in capital letters), or any other defined Windows Help keyword and press Shift+F1 to call up the Help Topic(s) on that item.

Context-Sensitive Help

Highlight any menu item and press F1 (or Shift+F1 or Ctrl+F1) to go directly to the Help topic(s) for that menu item. See also the Context-Sensitive Help menu item below.

[Help] Contents F1

The Contents menu option provides a list of main topics in the on-line User's Manual available in On-line Help, to help you understand various aspects of WinCALIS Author. The gray Search button at the top of the Help window is particularly useful. You can search for any keywords or topics using the Search feature and Help will find it and take you right to it. Click on "Full Table of Contents" to open a separate Table of Contents window. Or press the "Glossary" button for a glossary of Windows and WinCALIS terms, or the "Index" button for a book-style index to the on-line User's Manual.

[Help] Index

Opens WinCALIS Author On-line Help and takes you directly to the Index.

[Help] Search for Help on...

Opens WinCALIS Author On-line Help and takes you directly to the Search dialog box.

[Help] Context-Sensitive Help Ctrl+F1



Turns the mouse pointer into a Question-mark context-sensitive help pointer. You can then click with the Question-mark mouse pointer on any keyword in a WinCALIS command, task option, special response, or global variable keyword or on any pull-down menu item for which information is desired.

[Help] Keyboard Help

Keyboard Help displays on screen a picture of the current keyboard layout, with color-coded accent overstrike keys and the letters which they affect.

See Also: Accents and Diacritics

[Help] About Author...

About Author shows the current version, date, and serial number of your copy of WinCALIS Author.

Chapter 3: Lesson Development

Overview

Directions, questions, and feedback form the basis for any WinCALIS lesson. With this in mind, we devote this entire section to the technical and pedagogical aspects of directions, questions, answers, and feedback. This is where the students show what they have learned, and whether or not they comprehend the discourse in question. It is also where you can direct them to the right answer, provide encouragement and confirmation, or give additional information.

Popup (Directions) Window

The best way to think of directions is as fastened to a group of questions. The student will initially have to view a set of directions in order to answer the questions that will follow. Directions are automatically displayed on the screen the first time the student begins a lesson; thereafter, the student can access the directions at anytime by pressing the F9 function key.

At standard VGA resolution the Popup (Directions) Window may accommodate from 1 to 31 lines without the need for scrolling, depending on the font size being used. At the default Medium font size, at standard VGA resolution it accommodates up to 17 lines without scrolling. If the directions text fills more than one screen, a vertical scroll bar located on the right-hand side will appear, enabling the reader to scroll through the entire text.

Text entered in the Popup (Directions) Window can be linked to one or more tasks. When text is entered in the Popup (Directions) Window, WinCALIS Author notifies the system that this set of directions pertain to this and all following tasks in the lesson, until a new set of directions is typed in a new Popup (Directions) Window or you have reached the end of the lesson. If new directions are entered in another task, the previous directions get replaced by the new information. This new information will then be the information that is displayed when the user is presented with the task or presses the F9 function key.

Directions can be incorporated into a lesson in two ways:

Method #1

To create directions using the pull-down menu, complete the following steps:

- Select the Script menu, then the menu item Script Start-up Directions. (Alt+S,D).
- Type your directions in the Script Start-up Directions dialog box. Then press the Done button.

Method #2

- Click on the Popup (Directions) Window in order to make it active. Begin typing the new directions.

When entering directions for the first time, we recommend you use Method #1. Any additional directions you decide to enter must follow Method #2. Remember, the last set of directions entered are the ones displayed until a new set of directions is entered, or the student has reached the end of the lesson.

Note: Directions written via Method #1 in the Script Start-up Directions dialog box are placed by WinCALIS Author in the #0 task, when generating the *.CAL script file (see CALIScript). If there are no other directions written in the Popup (Directions) Window for Task 1, directions placed here will appear no different to the student from those placed in the Popup (Directions) Window for Task 1 (Method #2). However, if directions are placed both in the Script Start-up Directions dialog box and in the Popup (Directions) Window for Task 1, the directions for Task 1 will appear immediately after the Start-up Directions, replacing the start-up directions in the Popup (Directions) Window from that point on.

Interactive (Question) Window

WinCALIS Author allows you to ask four familiar types of questions: short answer, multiple choice, true/false, and fill-in-the-blank type exercises (in Cloze exercises). As stated in the preceding section, it is best to think of questions as being written on index cards in a pile. When you write the lesson, you deal with only one question at a time, as if you were writing the question on one side of the card, and turning it over to provide answers and feedback on the flip side. As you create more and more questions, they pile up like a stack of cards that you flip through to answer.

Methodology

As a rule, individual question development proceeds in seven steps, four of which (1,5,6 & 7) are optional:

1. Enter directions that apply to the Task.
- 2. Enter the question in the Interactive (Question) Window.**
- 3. Select the type of task from the Task menu option.**
- 4. Provide a list of anticipated responses with at least one Correct/Continue response.**
5. Provide feedback for each of the anticipated responses.
6. Select additional options.
7. Insert video, graphics and/or audio.

The Anticipated Response/Feedback Dialog Box

When you select any of the task types from the main Task menu, a dialog box will appear. The dialog box will have two text boxes, which will be used to supply responses and feedback; a set of option buttons on the left side of the dialog box to define response types; and a set of command buttons on the right side of the dialog box which will be used to move forward and backward through the anticipated responses and set various options. The dialog box features are described in the following sections.

Entering a Correct/Continue Response

The first anticipated response(s) should generally be the correct one(s), which can be indicated by clicking the small Correct/Continue option button located on the Response Type option box on the left side of the dialog box.

When you are finished typing the Correct/Continue response, you may click the mouse in the “Feedback [optional]” text box or simply press the Tab key to move the cursor to the Feedback text box. You may then type some words of encouragement, list any other Correct/Continue answers—or say anything you want. If you indicate no feedback, WinCALIS will supply the default feedback “Correct.”

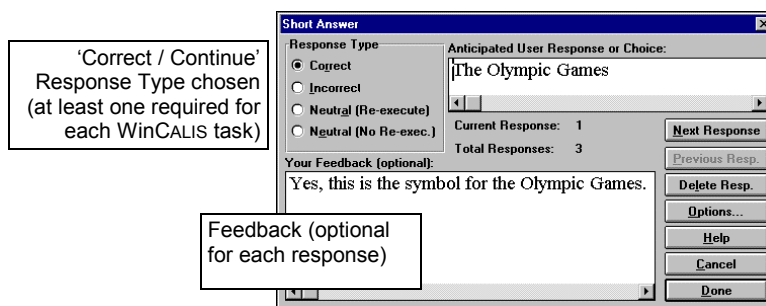


Figure 7: Response Dialog Box

When you are finished with the feedback, click on the Next button, and the Anticipated Response box will be cleared, in preparation for the next anticipated response. The Current Resp... and Total Resp... located on the top right side of the dialog box will display the current and total number of responses attached to the question so far. In the next two sections, you will learn how to use “wild card” characters to anticipate a wide range of responses with only a minimum of keystrokes and guesswork on your part. For now, enter as many individual Correct/Continue answers as you like.

Note: Every task the student has to complete must have at least one Correct/Continue answer. If the student eventually “gives up” on the question, WinCALIS will display useful feedback. Unless the special <GIVEUP> response is present (discussed in the section “Special Responses” in the section CALIScript), the first Correct/Continue answer will be shown to the student. By default, WinCALIS will move on to the next question when the student chooses any of the available Correct/Continue answers. It is possible to require students to answer all of the available Correct/Continue answers before moving on. This is discussed in the section “Task Options and Defaults.”

Entering an Incorrect/Remain Here Response

The Anticipated Incorrect/Remain Here Response is distinguished by clicking the small Incorrect/Remain Here option button that is located in the Response Type option box on the left side of the dialog box.

Anticipated Incorrect/Remain Here responses allow the author to provide helpful clues, review or reminders to a student who is on the wrong track. Therefore, in order to anticipate Incorrect/Remain Here answers, you must think of common errors which might be made in response to the given question. For example, if you asked for the first-person singular past tense for the verb “take,” you might anticipate the Incorrect/Remain Here response “taked.” Optionally, you can place the cursor in the “Feedback [optional]” text box and enter some feedback to tell the student the error and perhaps provide a clue to the answer. In the “taked” example, you might say, “Remember, 'to take' is an irregular verb.” When you are finished with the feedback for the anticipated response, click on Next to proceed to the next anticipated response or Done to move on to the next question.

Note: When you are adding new tasks to a lesson, WinCALIS Author saves you time by automatically take you to the Interactive (Question) Window for a new task. But when you are editing an existing task, it will take you back to the Interactive (Question) Window for the same task.

Entering a Neutral Re-execute Response

The Anticipated Neutral Re-execute Response is distinguished by clicking the small Neutral Re-execute option button that is located in the Response Type option box on the left side of the dialog box. In order to anticipate neutral answers, you must think of common errors which might be permitted in other instances, but are not relevant for the given question. For example, if you asked for the first-person singular past tense for the verb “take,” you might anticipate the response, “taken.” Optionally, you can place the cursor in the “Feedback [optional]” text box and enter some feedback to tell the student of the error, and perhaps provide a clue to the answer. In the “taken” example, you might say, “Remember, ‘to take’ is an irregular verb.” When you are finished with the feedback for the anticipated response, click on Next to proceed to the next anticipated response or Done in order to move on to the next question.

In other words, the Neutral Re-execute Response statement marker (=) is used in a WinCALIS task for student responses which are considered to be neither correct nor incorrect. Unlike an Incorrect/Remain Here response, a neutral response is not counted in a student's score.

This makes a neutral response ideal for making options available to students such as getting help for the current question or making a selection from a menu. Such choices are neither correct nor incorrect, and therefore do not receive automatic feedback.

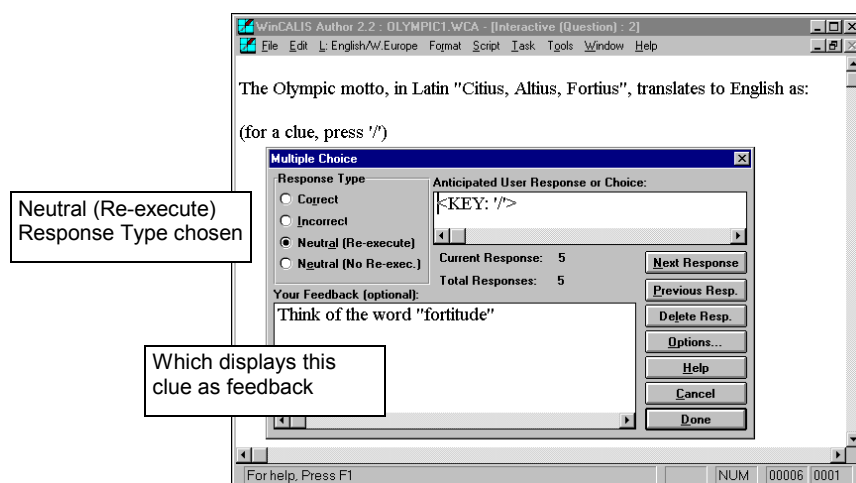


Figure 8: Neutral Re-Execute as a ‘clue’ task

Note: “Re-execute” refers to re-executing the task, WinCALIS’s normal behavior. What distinguishes the Neutral Re-execute Response from the Neutral No Re-exec Response is that in the case of the Neutral Re-execute Response, the entire task is *re-executed* after the response—re-executing every statement and command—just as it is in the case of Incorrect/Remain Here responses. The Neutral No Re-exec Response, on the other hand, does not lead to re-executing the entire task. After displaying any feedback or performing any of the commands contained in the feedback portion of the Neutral No Re-exec Response statement, WinCALIS does nothing, but simply waits for the student to respond again.

Entering a Neutral No Re-execute Response

The Neutral No Re-exec Response is selected by clicking the Neutral NOREP option button that is located in the Response Type option box on the left side of the dialog box. Like the Neutral Re-execute Response, the Neutral No Re-execute Response statement is used for student responses which are considered to be neither correct nor incorrect.

What distinguishes the neutral no-re-execute response however, is that instead of redisplaying the task to the student, WinCALIS will not rewrite the information in its windows. Therefore, this response is normally used in conjunction with audio or video for short answers. For example, as part of a task, you

may want to insert a particular word or graphic a student can click on in order to hear an audio clip relating to the question. In order to do this, you must insert a special response referring to the location of a particular graphic or word (creating a *hotspot*) in the anticipated response text box, and add the necessary commands in the “Feedback [optional]” text box to activate the graphic or audio. The student will then be able to click on the hotspot in order to display this special information. Once you are finished with the feedback for the anticipated response, click Next. If you are finished defining all your responses and wish to move on to the next question, click Done. When you click Done, the question, anticipated responses, and feedback become part of the lesson.

Selecting a Command Button in the Dialog Box

The Command Buttons on the right side of the Anticipated Response text box represent the following.

- The Next button will move you forward to the next anticipated response or to a new window where you can enter another anticipated response.
- The Previous button will move you backward to the previous response.
- The Delete button will delete the anticipated response and its corresponding feedback from the question.
- The Options button presents you with a host of options regarding feedback and error analysis. These options are explained in detail in Script (Default) and Task Options.
- The Help button will pop up on-line help dealing with the Feedback and Answers dialog box.
- The Cancel button will ignore any changes made in anticipated responses and feedback since opening the dialog box and return you to the Interactive (Question) Window.
- The Done button saves all contents of the dialog box and moves you to the next question (add task mode) or the same question (edit task mode).

You must enter each anticipated response fully and exactly until you learn how to use the wild card characters presented in It's All in the Wild Cards .

Note: If you are working with two or more language keyboards, you can use the F3 function key to toggle between the currently selected language keyboard and the “top” language for your workstation (by default English/W.Europe) in the L: [Language] menu.

Cutting, Copying, and Pasting Text in Dialog Boxes

You will sometimes need to copy the feedback for one response into the Feedback box for another response, or use the same feedback for a whole series of responses. In this situation you would first highlight the text you wished to copy with the mouse or keyboard arrow keys. Then you would use the Windows Copy and Paste commands. However, in certain types of Windows dialog boxes, including the Anticipated Response/Feedback Dialog Box, the pull-down menus are dimmed and not available for standard editing operations such as cutting, copying, and pasting text.

Fortunately, this does not mean that these operations are not possible there! Highlight the text to be cut or copied as you normally would. Then just use keyboard shortcut keys for each operation. WinCALIS Author supports *both* sets of standard Windows shortcut keys for these operations:

- *Cut:* **Ctrl+X** or **Shift+Delete**
- *Copy:* **Ctrl+C** or **Ctrl+Insert**
- *Paste:* **Ctrl+V** or **Shift+Insert**

Shortcut keys are usually shown just to the right of menu commands in all Windows pull-down menus. So if you forget any of these shortcut keys, just pull down the Edit menu and you will see which key to use for the editing operation you need.

What's your Question Type?

Many question types, including the Flash Card variation of the Short Answer and the True/False question, rely on a single anticipated response and use default feedback, so the process of creating tasks remains short and easy. Several factors must be considered when you decide what type of question to ask the student. These are the same factors you consider when you make up any kind of test: the learning objective, nature of the text passage, ability level of the student, etc. Fortunately, a lesson may consist of any combination of the question types, so you may give comprehensive treatment to the learning assignment.

We will now examine each of the four question types in order to see how the process is implemented for each one.

Note: The audio and video selections are covered in the section Multimedia in WinCALIS.

Under the Task menu you will find the following question types:

Short Answer
Multiple Choice
True/False
Cloze Exercise

There are also the Flash Card and Presentation variations of the Short Answer.

Short Answer

When you select the Short Answer type, by clicking on the Task menu and selecting the menu item Short Answer (Alt+T,S or Ctrl+S), a Short Answer dialog box will appear. The dialog box will have two text boxes, a set of option buttons on the left side and a set of command buttons on the right side, which will be used to anticipate responses and select options, as described earlier under the heading "The Anticipated Response/Feedback Dialog Box." The cursor will appear in the Anticipated Response box, ready for you to enter an anticipated response to the question.

If you wish to change either the feedback or anticipated responses associated with a particular question, click on the Next or Previous button until you find the response and/or feedback. Make the necessary adjustments then click on Done to save the changes for that question.

Flash Card

The Flash Card question type is just a variation of the Short Answer question. When you want to create a Flash Card question, you will select the Short Answer option by clicking the Task main menu option and selecting menu item Short Answer (Alt+T,S or Ctrl+S).

The flash card question, like the Short Answer question, is a short method at getting a bit of information from the student without anticipating responses, providing feedback, or doing error analysis. This type of exercise lends itself very nicely to vocabulary drills and similar rapid one-word queries.

Simply enter any number of Correct/Continue, Incorrect/Remain Here, and neutral responses in the Flash Card dialog box. Each response should be followed by clicking the Next button. If you are working with two or more languages you can use the F3 function key to toggle between the currently selected language and the “top” language keyboard for your workstation. When you are finished, click Done.

Multiple Choice

The Multiple Choice type of question functions in the same manner as the Short Answer. But in the case of Multiple Choice, the Interactive (Question) Window screen is automatically formatted in a vertical column of choices for you. Each item typed in the Anticipated Response box becomes an available choice. There must always be at least two choices given, one of which is correct. From the Task menu select the menu item Multiple Choice (Alt+T,M or Ctrl+M).

A Multiple Choice dialog box much like the Short Answer one will appear, with two text boxes and option buttons on the left and right, the cursor blinking in the Anticipated Response box (actually the “Available Choice” box) and the option button set to Correct/Continue in the Response Type box, ready for you to type the text for an available choice. (For how to adjust the placement of choices on the screen, or even how to have a choice span more than one line, see the Tab Stop in Multiple Choice option in the More Options dialog box.) After typing the Correct/Continue choice and any feedback you would like, you may move on to indicate some Incorrect/Remain Here choices. Use the mouse or the Tab key to move to the Response Type box and choose Incorrect/Remain Here. Any feedback is optional. WinCALIS Author takes care of providing the heading letters (a-z in Roman script) from which the student will choose and randomize the sequence of the options given. When you are finished with the question, click Done.

Note: WinCALIS Author randomizes the choices by default, when it generates the *.CAL file from the *.WCA Author database file, saving you the need for human randomizing of Correct/Continue answers. You may deselect this feature in the Script Options or Task Options dialog boxes, if the order of choices is important. If you make use of the convenient automatic randomization, you should avoid choices like “None of the above,” using instead a phrase like “No correct answers” or “None of the choices is correct.”

True/False

There is no easier WinCALIS question to produce. When you are finished typing the statement whose truth is to be evaluated in the Interactive (Question) Window, simply click on the Task menu, then select the menu item True/False, TTrue... (Alt+T,T or Ctrl+T) or True/False, False... (Alt+T,Z or Ctrl+Z), whichever is correct for the question. The feedback is optional. Authors may wish to use alternative words rather than the default words “True” and “False,” for example, “Correct” and “Incorrect” (or their equivalents in another language). You do this simply by deleting “True” and “False” and retyping the alternative words.

In the True/False dialog box, the Anticipated Response and “Feedback [optional]” boxes behave much as in the previous question types, except that the Response Type button and the Delete command have been disabled. When you are finished with the question, click on Done.

The Interactive (Question) Window will be automatically formatted for you, with the “True” response and the “False” response arranged to the left and right horizontally, under the true-or-false statement. Note that with a little forethought, this automatic formatting can lend itself to other types of either/or tasks as well. For example, with the No Score and No Praise options selected, you could use the “True/False, False” task type for a neutral question such as the following:

“Do you wish to watch the demonstration again or continue on?”

For the False (Correct/Continue) response, you could type something like “Continue on” and for the True (Incorrect/Remain Here) response, you could type something like “Watch it again.” The screen

would be automatically formatted like this, with each item responsive to either a mouse click or a key response:

Watch it again.

Continue on.

Note: For both Multiple Choice and True/False task types, WinCALIS Author uses the first letter in the Anticipated Response text box to generate a special keyboard response, making it possible for the student to respond by pressing the key corresponding to that letter without pressing ENTER. Author permits you to type any valid Unicode character in that box, but the first character should preferably be one which appears on the keyboard, to avoid unexpected results. Be creative. In some languages, “O” symbolizes “true,” while “X” symbolizes “false.” Thus “O” and “X” can be used as the first letter each line, followed by a space and then a word in the appropriate language.

Cloze Exercise

In the Cloze question type, students are presented with a passage containing empty blanks for missing words or phrases. These blanks are to be filled in by the student, who must rely on familiarity with the passage or contextual and noncontextual clues.

The construction of the Cloze exercise proceeds much like that of the other types of exercises, but there are no questions in the usual sense. Essentially, each question is: “What is the answer that goes in this blank?” You merely type in a text passage in the Interactive (Question) Window and tell WinCALIS which words to blank-out, (i.e. the question answers) in a process called “marking.” You also have the option of “marking” clues or typing in noncontextual clues in the Noncontextual Clue text box that will be displayed if the student requests help.

After you have completed entering the text passage, you must make sure that it is exactly as you wish, because once you begin marking, it is laborious to alter the text. Therefore, we strongly suggest that you copy and paste the text from the Interactive (Question) Window into the Scratchpad Window, and save it under the Document file option with a different filename. That way it will always be retrievable in its original form, and may be copied back to the Interactive (Question) Window easily if, for some reason, difficulties arise while you are marking the text and you wish to restart the marking process. When you are certain that the passage is acceptable in the Interactive (Question) Window, you then mark the Cloze task. Specific descriptions of the various parts of a Cloze exercise are described in the following sections.

Tip: WinCALIS Author will automatically leave a blank line at the top of the Interactive (Question) window containing the cloze text, when it generates the script. If you wish your cloze text to begin at the very top line, insert a <BLANK> command at the beginning of the first line of your text. This will replace the blank line with your text. (Don't enter a hard return after the <BLANK> command, or you will undo its effect!)

Formatting Your Cloze Exercise

From the Task menu option choose menu item Cloze Exercise... (Alt+T,Z or Ctrl+Z). A Cloze Exercise dialog box will appear with three text boxes and a set of command buttons to the right side of the dialog box. The topmost text box Question Text contains the text typed prior to opening the Cloze Exercise dialog box. You mark text in order to create blanks for students to fill in. The procedure for marking text is simple, although as mentioned earlier, very difficult to undo. Simply use the mouse to highlight the word or words which will constitute the first answer. Then enter the marking phase by selecting the Mark command button on the right of the dialog box. An information box with a warning informing you that the text can not be reedited once you are in the marking phase will appear. Click on the Yes button. This will mark the highlighted text as the first answer and tell WinCALIS to blank out the answer when the passage is displayed for the user. The answer will be enclosed in double-angled brackets, like this: This is the <<answer>>.

At this point, you have the option of marking a contextual clue for the first answer or moving ahead and highlighting more answers (with optional associated clues).

Clues

When the student does not know the answer in a Cloze exercise, he or she may ask for a clue by entering the question mark (?) instead of the answer. Two types of clues are available: contextual and noncontextual. They operate as follows.

Noncontextual Clue

Noncontextual clues are informational sentences or phrases that you type in the “Noncontextual Clue” text box just before you begin marking the word selected as the question the student needs to type. The clue entered is linked with the question passage typed in the Interactive (Question) Window. This clue should give a noncontextual hint. Only one clue is allowed for each marked answer.

Contextual Clue

Contextual Clues are segments within the question passage itself that will show up as highlighted when requested by the student. For example, if you wish the student to fill in the blank at the end of a sentence with a verb, you may make the noun in the sentence appear highlighted when the student requests a clue (hence the word Contextual). You may mark contextual clues in the same way you mark answers. Each clue is associated with the most recently marked answer.

When the student enters a “?” the default clue feedback will appear and the marked clue will change color in the question text.

Additional Anticipated Cloze Responses

You may add additional Correct/Continue or Incorrect/Remain Here anticipated responses to any word that has been previously marked. This is accomplished by highlighting the already marked word and clicking on the command button More. A More Answers... dialog box similar to the dialog boxes in short answer, multiple choice etc. will appear. The cursor will appear in the Anticipated Response text box, ready for you to enter an anticipated response. You can enter as many Correct/Continue or Incorrect/Remain Here anticipated responses as you wish. When you have finished, click on the command button Done to save the content for that particular marked word. Repeat the process for any additional or new words for which you want to add Anticipated Responses.

The standard wording for the default clue feedback is “Look at the highlighted text for a clue” and the default color for clue highlighting is red. You may change the default clue feedback globally or on a question to question basis. To change the clue feedback globally choose Script Options under the Script main menu. In order to change the clue feedback just for the task you are working on, choose Options in the Cloze Exercise dialog box. In the Cloze Options dialog box which appears again choose the Options button. A Task Options dialog box will appear which looks the same as the one which changes the global defaults. However, when accessed in this manner, it will only change the task you are working on.

You may change the default wording for Correct/Continue and Incorrect/Remain Here feedback in much the same way. These changes can apply either globally or locally depending on where the changes were made. Change the text in either the first text box, Default Correct/Continue Feedback, or the third text box, Default Incorrect/Remain Here Feedback.

Cloze Options

Clues, Answer Marking, and other Options

WinCALIS Author provides an automatic means for determining color and type attributes for clues, filled-in answers, and blanking (inserted in the text before the Correct/Continue answer is filled in). You may change the default either by going into the Cloze Options menu item under the Script menu in order to make global changes, or by changing the default directly when you are in a Cloze task (under the Options button). If you choose the latter method, changes will apply only to the individual Cloze task on which you are currently working. This dialog box controls several options which are now described.

Emphasize Answer

You may choose from eight colors for these answers: black, blue, green, white, yellow, purple, cyan or red from the option buttons. You may also choose from four type styles: **bold**, *italic*, underline, and ~~strikeout~~ from the Emphasize Answer option check box.

Emphasize Clue

To emphasize clues you may choose from eight colors: black, blue, green, white, yellow, purple, cyan and red from the option buttons. You may also choose from four type styles: **bold**, *italic*, underline, and ~~strikeout~~ from the Emphasize Answer check box.

Enable Fixed Blank Length

At times the length of the blanked words can easily give away the answer to a particular question. Therefore it may be desirable to control the length of the blanks in the text dialog, thereby discouraging the student from guessing the correct answer based on size. This is accomplished by clicking in the Enable Fixed Blank Length check box. WinCALIS Author will automatically evaluate and select the optimum size that is allowed in the Cloze exercise and display the size in the Fixed Blank Length text box. The size is equal to the largest marked word in the text from the dialog box. If however the length is inappropriate for your script, you may at any point change the size of the space in the script. If you enter a space length smaller than the largest anticipated response, a warning message will appear, and WinCALIS will automatically fill in the size to match the largest marked word in the text.

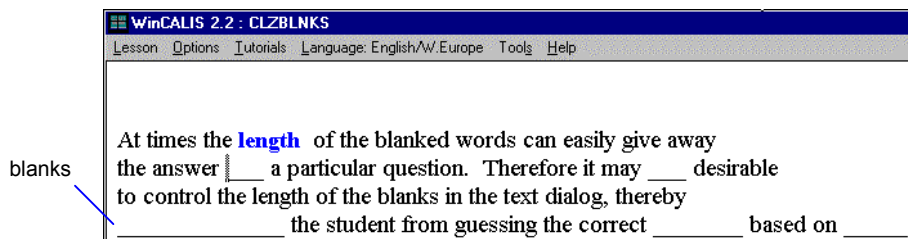


Figure 9: Cloze Exercise with default settings

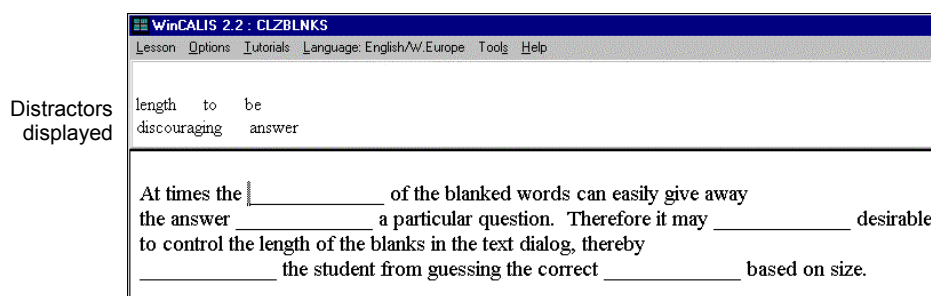


Figure 10 Cloze Exercise with options:

**Fixed Blank Length: enabled
(based on length of longest blank)
Display Distractors: enabled
Number of Distractors per line: 3**

Display Distractors

Distractors can serve as a powerful student aid tool, especially at the elementary level. You can enable this option by simply clicking on the Display Distractors check box. The distractors will appear on the text box labeled Distractors in WinCALIS Author, and in the Presentation (Text) Window to the student in WinCALIS. If you add additional distractors after the initial first marking you must refresh the screen in order to update the text box so you can view any new additions you might have made. This is done by clicking on the command button Distractors, when you are in the Cloze Exercise dialog box.

Number of Distractors Per Line

By default WinCALIS Author will allow seven distractors per line; however at times the distractors may be longer than what has been anticipated. In these cases some distractors will only be partially displayed on the screen or might even be completely off of the screen. To avoid this, we can change the number of distractors that appear in one line simply by changing the default in the Number of Distractors per line text box to an appropriate size.

Randomize Distractors

Randomizing the distractors is another enhancement that can be incorporated into your scripts. This is accomplished by clicking on the Randomize Distractors check box.

Mouse Activated Distractors

If general comprehension of the text is of greater priority than the orthography of the language to be learned, particularly at the initial stage of language learning, then enabling the distractors to be mouse-activated can be of assistance to the student. With this feature enabled, a student user needs only to click on a distractor to input that as a response (WinCALIS will then perform its answer analysis).

Enable Marking of Punctuation

By default WinCALIS ignores punctuation at the end of a word for cloze exercises. However, there are instances where punctuation is an essential part of language learning. Therefore WinCALIS allows you the option of determining when to validate punctuation. This option can be made active by simply checking the Enable Marking of Punctuation box. Note that if this box is not checked, punctuation will be ignored in answer determination.

All of the above Cloze options can be changed in one of two ways. Either select the Cloze Options menu item under Script to make global changes (changes that will apply to all Cloze questions within the script), or select the Options command in the Cloze exercise dialog box to change individual questions. If you choose the latter method, changes will apply only to the individual task on which you are currently working.

Moving Among Tasks with the Task Menu

In keeping with the analogy of the questions in the task list as a stack of index cards, you should remember that you produce and flip through questions in the order that you write them. You do not deal with all the questions at once; rather, you deal with each question as a separate component of the lesson.

Next/Previous

To this end, the Next and Previous menu items under the Task menu will move you through the pile of questions, with respect to your current place in the question pile. The shortcut keys for Next Task are Ctrl+N or the gray + key; the shortcut keys for Previous Task are Ctrl+P or the gray - key.

Delete

The Delete menu item under the Task menu (or the shortcut key Ctrl+D) allows you to permanently remove the current question from the pile, after first verifying that you want to do so. WinCALIS Author will then link the questions surrounding the deleted question together such that they will become consecutive.

Go to Task

The Go to Task menu item under the Task menu (or the shortcut key Ctrl+G) will present you with a text box where you give the task number you wish to move to. If a task number is entered that is greater than the total number of tasks, a warning message and the number of the last task created in the script will appear on the screen. This gives you a handy way to jump to the last task in your script.

Rearrange Tasks

The Rearrange Tasks menu item under the Task menu (or the shortcut key Ctrl+A) will enable you to manipulate tasks. (See *Authoring Basics*, *The Task Menu*, *Rearrange Tasks* for more information.)

See Also: What's your Question Type?
Task Menu

Question Sequencing in WinCALIS

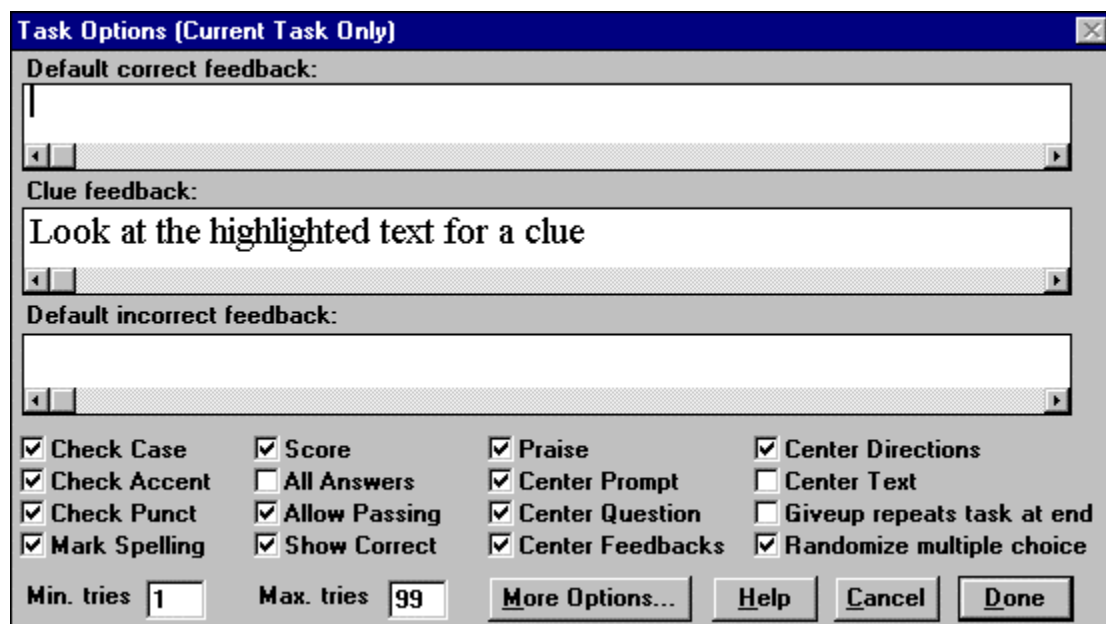
The order in which the questions appear when actually run within WinCALIS is of obvious importance in lesson development because top-down and step-wise questioning techniques depend largely on question ordering. Therefore, you must understand the relationship between how questions are entered in the WinCALIS Author Interactive (Question) Window and how they are played out in the WinCALIS Interactive (Question) Window.

As stated above, WinCALIS Author considers the questions to be tied in a loop with the last question entered connected to the first question in a given script. When Random task order is left unchecked in the Script Generation Options dialog box, WinCALIS will ask questions sequentially, beginning with the first question created in WinCALIS Author.

Global (entire script) and Task Options

You have complete control over several options which go into the formation of an overall communication protocol. By customizing the script and question generation options, you determine what and how things are conveyed to the user.

You may choose to apply these options globally (affecting every task in the whole script), or to an individual task. The way to apply options globally is by clicking on the Script menu option and choosing the menu item Scripts Options. The method for applying task-by-task options is by clicking on the Options button when you are in the Task dialog box for that particular task.



The Default Correct/Continue, Default Incorrect/Remain Here and Clue Feedback boxes supply the user with some standard feedback in lieu of specific feedback for each situation. When you enter a message in any of these boxes, it will be displayed as the feedback for Correct/Continue, Incorrect/Remain Here, or clue information when there is no specialized feedback provided. For instance, instead of just “Correct,” you could make the default Correct/Continue feedback “That's right! Keep up the good work.”

Note: The Script Options dialog box, which is opened by selecting Script Options under the Script menu, is identical in function and appearance to the Task Options dialog box, which is opened by clicking on the Options button in the answers/feedback dialog box for each task type under the Task menu--except for the fact that the script options under the Script menu are in effect for every task in a script (global), while the task options in the answers/feedback dialog boxes under the Task menu are in effect only for the current task (local), overriding any conflicting global Script Options.

The following options pertain mainly to the actual scoring of the user's responses and how WinCALIS will allow the user to proceed in answering the questions:

Check Case

Check Case refers to the capitalization of the words in the response. When this item is selected, any answer which does not match a Correct/Continue answer precisely in terms of upper- and lower-case letters will be counted as Incorrect/Remain Here.

Check Accents/Punctuation/Spelling

Like Check Case, Check Accents and Check Punctuation must have exact matches in each of their respective characteristics to be scored correct. The most stringent grading would then be to select both of these boxes, along with the Check Case box, but each characteristic may be selected independently of the others.

Mark Spelling

The Mark Spelling option enables “Spellmarking,” in which the student answer is automatically marked up, showing differences between it and the first or closest-match Correct/Continue answer.

Checking the Score

The Score option tells WinCALIS Author to count that particular question towards the student's overall score.

All Answers

The All Answers option requires the student to furnish every Correct/Continue answer that you have listed, before the question is considered completed and correct.

Allow Passing

The Allow Passing option gives the student the opportunity to pass a question after the minimum number of attempts specified by Min. Tries. When a user gives up on a question, the answer may or may not be displayed, depending upon whether you've selected Show Correct/Continue or not.

Show Correct

When you check Show Correct, you allow WinCALIS to display the Correct/Continue answer(s) you have specified for the question after the student gives up or attempts the question the maximum number of times indicated in Max. Tries.

Praise

The Praise option enables feedback for the Correct/Continue and anticipated Incorrect/Remain Here answers to a question. If Praise is not selected, all feedback is suppressed.

Center Options

The Center Prompt, Center Question, Center Feedback, Center Directions, and Center Text selections will display all lines centered in the specified windows. The default setting for each of these options is centered (checked). If unchecked, lines are displayed left-justified.

See Also: CENTERing a Text Line

Giveup repeats task at end

If the student eventually gives up on a question, WinCALIS will display the answer to the student and move on to the next question. When the student has answered all the questions with this option enabled, WinCALIS will return to those questions the student gave up on and proceed to ask the question(s) once more. This process will continue until all questions in a lesson have been responded to correctly.

Randomize Multiple Choice

The Randomize Multiple Choice selection randomizes the displayed choices (the anticipated answers) in the multiple choice task type only.

Max. Tries

Max. Tries is the maximum number of attempts the student is allowed to make before WinCALIS considers the question answered incorrectly. It will display the Correct/Continue answer if you have selected Show Correct.

Min. Tries

Min. Tries is the minimum number of attempts that the user must make before giving up on the question.

More Options

The More Options button in the Script Options and Task Options dialog boxes opens another dialog box presenting you with additional options.

More Task Options (Current Task Only)

<input type="checkbox"/> Allow Repeat	<input checked="" type="checkbox"/> Use D Window	<input checked="" type="checkbox"/> Blank Question Window
<input checked="" type="checkbox"/> Allow Giving up	<input type="checkbox"/> Disable Keyboard	<input checked="" type="checkbox"/> Blank Text Window
<input checked="" type="checkbox"/> Pass between groups	<input type="checkbox"/> No Mouse Hand Pointer	<input checked="" type="checkbox"/> Blank Directions Window
<input type="checkbox"/> All n Answers	<input type="checkbox"/> Limit Answer Length	<input type="checkbox"/> Quiz

Enter number of answers n: Question window top line:

Enter answer length n: Left margin column number:

Tab Stop in Multiple Choice:

Note: The More Script Options dialog box, which is opened by selecting Script Options under the Script menu, then clicking on the More Options button, is identical in function and appearance to the More Task Options dialog box, which is opened by clicking on the Options button in the answers/feedback dialog box for each task type under the Task menu, then clicking on the More Options button--except for the fact that the script options under the Script menu are in effect for every task in a script (global), while the task options in the answers/feedback dialog boxes under the Task menu are in effect only for the current task (local), overriding any conflicting global Script Options.

Allow Repeat

Although there is little point in asking a student a question already answered correctly, it may be useful to allow tasks that present information, or allow students to make choices, to be repeated several times. This check box option provides the ability to control whether or not a task may be repeated.

See Also: Allowing Repetition of Completed or Deleted Tasks

Allow Giving Up

The Allow Giving Up option by default allows the student who has made the minimum number of attempts at a question to press the giveup key [F6] and view the Correct/Continue answer. WinCALIS displays the first Correct/Continue answer listed for that task in a pop-up window and moves on to the next question. You can disable or enable this feature by clicking the check box Allow Giving Up.

By default, when a student completes a task correctly or gives up, the task is marked Correct/Continue and placed on a list of tasks which have been completed.

Pass Between Groups

As mentioned earlier in this section, tasks which appear under a common set of directions are grouped together. In general, students can pass between tasks, with the exception of a Cloze task. You can prevent the student from passing by deselecting the Pass Between Groups check box.

See Also: Group Behavior

All n Answers

The All n Answers option requires the student to match more than one Correct/Continue answer, one at a time, but fewer than the total number of Correct/Continue answers. The number of answers the student must enter out of the total number of Correct/Continue answers can be specified by entering a number in the Enter number of answers n: text box. Compare the All Answers Option.

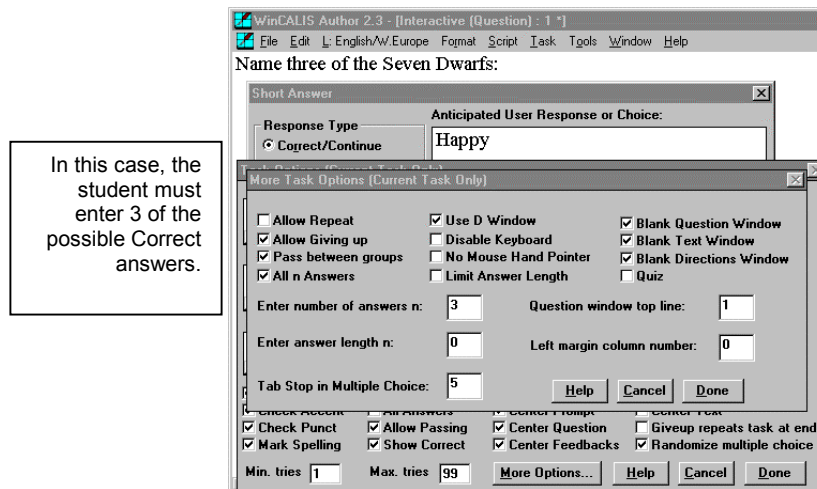


Figure 11: All n Answers

See Also: Requiring Students to Match One or All Correct/Continue Answer Possibilities

Enter Number of Answers n:

This text box is used in conjunction with the All n Answers box to control the number of Correct/Continue answers you want the student to enter before continuing to the next task.

Use D Window

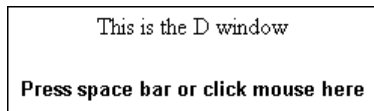


Figure 12: Feedback appears in the D window

With this check box marked, WinCALIS Author automatically sends feedback to student responses to the Diagnostic Feedback Window. If you deselect the “Use D Window” option, the feedback is left in the Interactive (Question) Window.

See Also: Sending Feedback to the D Window

Disable Keyboard

The WinCALIS author default allows the student to use the keyboard as a means of input. By selecting the Disable Keyboard option WinCALIS allows authors to disable keyboard input in cases where the student response should be given by using the mouse.

No Mouse Hand Pointer

The mouse pointer (mouse cursor) in WinCALIS is normally an arrow. It turns into a hand pointer when it is over a “hotspot” hyperlink on the screen. WinCALIS allows authors to keep the arrow pointer arrow (instead of turning into a hand pointer) in cases where the hand pointer is undesirable, such as when the student is asked to click on a single Correct/Continue hotspot in text or in a graphic on the screen.

Limit Answer Length

This option sets a limit on the amount of characters a student can type on the screen. When the student types more characters than necessary, WinCALIS beeps and shows the message “Your answer is too long.” The acceptable length is determined by WinCALIS to be the longest possible anticipated Correct/Continue response given in that task or the number of characters indicated in the Enter answer length n: text box.

See Also: Limiting the Number of Characters the Student can Type On-screen

Enter Answer Length n:

This option sets a limit on the number of characters the student can type on the screen.

Tab Stop in Multiple Choice:

This option sets the amount of space WinCALIS Author will leave to the left of the column of choices in formatting the multiple choice question type. The default tab stop is 5. If the text for any choice is unusually long, you may leave more room for the text by decreasing the tab stop.

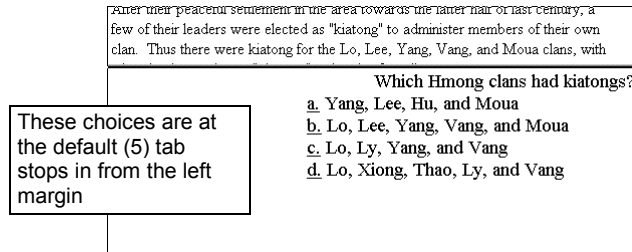


Figure 13: Tab Stop in Multiple Choice

Tip for advanced authors: If you wish to have a multiple choice item span more than one line, you can edit the *.CAL script to achieve this. But you can also do this in WinCALIS Author and have Author automatically format it for you with the following trick. Preformat your text in the Scratchpad or any other window in two or more lines with hard returns (in the same way that you would if you were creating the multiple choice by hand in the Scratchpad). E.g.,

```
choice text line 1
? [necessary number of tabs] choice text line 2
? [necessary number of tabs] choice text line 3
...
```

Then cut and *paste* the text into the Anticipated Response (Choice) text box, rather than typing it there. Your hard returns will be preserved when Author generates the script!

Blank [Window]

This option for the Interactive (Question) Window, the Presentation (Text) Window, and the Popup (Directions) Window clears the entire screen of old text when new text is entered into it, or the task is re-executed. Disabling this option will cause the text to be appended at the bottom of the previous text entered.

The syntax for this command is as follows:

<BLANK: Q> to clear the Interactive (Question) Window

<BLANK: T> to clear the Presentation (Text) Window

<BLANK: DIR> to clear the Popup (Directions) Window

See Also: Clearing the Interactive (Question) Window for Each Task

Quiz

This option turns on Quickie Quiz mode, requiring the student to enter name and Social Security Number (U.S.) or other identifying number, and records all questions and student responses in a file on disk.

See Also: Turning an Exercise into a Quickie Quiz

Interactive (Question) Window Top Line:

This option displays text in the Interactive (Question) Window beginning with the specified row. Previous rows are left blank as a top margin.

Left Margin Column Number:

This option displays text in the Interactive (Question) Window, the Presentation (Text) Window, and the Popup (Directions) Window beginning with the specified column.

Script Generation Options

The Script Generation Options dialog box will present you with additional script generation options that will affect the entire script globally. These options, available under the Script menu, function in the following way.

Note: Should you choose to effect global script options for all tasks, do this before entering any individual tasks to make changes on a case-by-case basis.

Random Task Order

This option will randomize the tasks within their groups. For example, if you had two groups of tasks, one on pronouns and the other on verbs, and each group contained five tasks, the randomize option would “mix up” all of the pronoun tasks, but still present all of them before any of the verb tasks. A group is usually indicated by a new Popup (Directions) window.

Student Chooses # of Tasks

The student will have the choice of attempting 10, 20, or 30 questions from within the given lesson. If there are fewer than ten questions in the lesson, the student will automatically be presented with all of them.

Use [English/W.Europe] Keyboard

When checked, this option sets the language keyboard automatically for the student to the language currently selected in WinCALIS Author. In addition, the student will automatically be asked if he wishes to display the virtual keyboard on screen. When not checked, the student language keyboard remains the top menu language, usually “English/W.Europe,” or whatever language keyboard was most recently selected during the current WinCALIS session. In either case, the student has the option of resetting the language keyboard using the Language menu.

Info Line in Task Windows

When checked, this option causes a message to appear in the title bar of a WinCALIS Author window listing the graphics, audio clues, and video clues associated with the particular task. This line serves as a reminder to you of the files associated with the script. If there are no graphics or audio or video clips associated with the task, then the title bar will not have such a message.

Center Text in the Window

When enabled, this option will center all text in the Lesson Description Window. The option to center text in other windows is found in the Script Options and Task Options dialog boxes.

Note: Script Generation Options are so named to distinguish them from ordinary CALIS Script Options (“defaults”) and Task Options. They are authoring options available to you, which result in the generation of various CALIS statements and commands in the *.CAL script to achieve the desired result. In contrast, when you select various CALIS options from the Script Options) and Task Options dialog boxes, Author inserts Options keywords in the script in a default statement or in individual task label statements.

Running a Lesson in WinCALIS

When you have completed a lesson to your satisfaction, having set all the script options as you like, you are ready to try out your lesson on yourself. Like a chef sampling your own cuisine, you will want to examine your creation before your students use it.

In order to run your WinCALIS Author lesson, you will first need to save your *.WCA author database file and generate the *.CAL script file, as explained in the section Authoring Basics. After starting WinCALIS, if the Lesson Window does not already show *.CAL files, click on the Options menu and choose Text Conversion (*.CAL) mode. Go to the directory in which you saved the lesson and you will find it in the alphabetized list of *.CAL suffixed lessons. Simply double-click on the lesson and it will be off and running. An automatic conversion will be made to the *.WCL format, so that the next time you run the lesson, if you haven't changed it, no conversion will be necessary.

You can also run a script directly by selecting the Run Script option under the Script menu.

When you are multitasking in Windows, moving between WinCALIS Author and WinCALIS, new or updated files are automatically may not be indicated in the WinCALIS Lesson Box. This is because WinCALIS will not automatically reread the current directory when you switch (back) to WinCALIS from WinCALIS Author. Therefore, you must access the drive to give a new directory reading so the changes will be reflected in your lesson box.

The quickest and easiest way to do this is to change the Conversions option from Text to None and back to Text so the drive will be accessed. If you do not follow this procedure newly created lessons will not appear in the WinCALIS Lesson Window.

The Value of Experimentation

You are now equipped with all the rudimentary knowledge necessary to make lessons. There is still plenty more to learn, but at this point, you should have a pretty good feel for what WinCALIS Author is all about. In the next section, you will learn more about customizing your screen aesthetics, your keyboard and your lesson so that a broader range of responses may be processed using wild cards.

But for now, we strongly suggest that you practice your fundamental authoring skills by assigning yourself a few lesson plans to implement. You might whip up a lesson on pronouns, or present a translation of "Little Red Riding Hood" and have your students figure out the vocabulary. You are limited only by your own imagination. Try something that allows you to apply and practice the technical knowledge you have acquired. As you know well, there is no better way to learn than by doing.

Chapter 4: Customizing and Communicating

Overview

In other sections you learned the authoring basics of WinCALIS. In this section our main focus is the menu options, and in particular the commands you can use in WinCALIS Author scripts to empower and refine your lessons. Sophisticated anticipated response handling (through wild cards), aesthetics, effective screen design, feedback flexibility, and tutorial help, along with extra-WinCALIS file use will be discussed.

It's All in the Wild Cards

Up to this point, we have learned the bare essentials for making live WinCALIS scripts, but not quite enough to make them very useful or practical. One step toward true applicability is taken now with the introduction of the wild card. Wild cards will enable you to anticipate a wide range of student responses with just a few keystrokes, thereby saving you the trouble of listing all possible answers verbatim. Wild cards are most often used in anticipating whole classes of Incorrect/Remain Here answers.

But first, we need to know exactly what a wild card is.

Definition

Just as with card games, a computer wild card is a character, or set of characters which can stand for other sets of characters. For example: the same way a joker card can stand for any other card in the deck, the ampersand character & can stand for any other character or group of characters in a student response. There are various wild cards, each functioning in its own special way to give you flexibility in anticipating answers.

Implementation

Since we have already mentioned the & wild card, the easiest and most flexible one, we will use it to illustrate wild card use in general, then explain the specifics of the other wild cards in the following section. The & wild card matches any sequence of letters, so if you enter & *moon* as an anticipated response, it would match anything ending in "moon" preceded by a space character, such as "the man in the moon", as well as whimsical phrases such as "the cow jumped over the moon". You can put the & wild card anywhere in the response and at that juncture, WinCALIS will accept any input.

Whenever you enter an anticipated response in the Answers/Feedback dialog box under each task type in the Task menu, you may insert wild cards (though the Short Answer task type is better-suited for the application of wild card characters for answer analysis than the others, for example: the Multiple Choice task type).

Anticipated responses include Correct/Continue, Incorrect/Remain Here, and neutral answers.

Later in this section, you will learn how to formulate anticipated responses which contain wild card characters which do not function like wild cards, e.g. being able to anticipate “Jack & Jill” where the “&” character is a literal “&” character and not a wild card.

The Wild Card Characters



Wild cards can be classified in terms of the amount of text which they match--what their bounds are. Some wild cards just match single characters. Other wild cards are always bounded by spaces--they can match anything within a word. Still other wild cards have no limitations--they will match anything at all. For the purposes of WinCALIS authoring, we define a word as a group of characters bounded by space characters (or the beginning or end of the anticipated student answer).

Wild Cards Which Match Single Characters

- ^ The caret wild card will match any single character. For example: the anticipated response, “^ate” will match “date,” “fate,” “late,” and nonsensical words like “xate” and “aate.” It will not match “grate” or “berate,” because more than one letter precedes the “ate” part.
- [] The bracket wild card will match any one of the letters found within the brackets. No comma or notation is needed in between the list of the characters, because the bracket notation tells WinCALIS to match only one character. For example: the anticipated response, “[blr]ump” will match “bump,” “lump,” and “rump,” but not “jump” or “clump,” etc.
- [-] In order to anticipate a long series of sequential characters, you may use the hyphen “-” to separate the bounds of the sequence within the brackets. For example “[A-Z]” will match any capital letter between “A” and “Z” (but not any lower case letters). “[1-9]” will match the numerals 1 through 9. Specifying the first and last letters of the Cyrillic alphabet will match any Cyrillic character. And specifying the first and last characters in the Unicode “Han” (Chinese-Japanese-Korean) character pages will match any Han character.

Note: The sequence itself is defined by the ASCII table, ANSI table, or Unicode table, which assign a code number (essentially the same one) to every character. See an ASCII or ANSI chart, contained in most general computer manuals, or the Windows “Character Map” Accessory to check on the sequence. For non-ANSI characters, you may also consult the Unicode reference manual, *The Unicode Standard, Version 1.1* Vols. 1 & 2. (Addison-Wesley, 1995?). Perhaps most conveniently of all, you can select the “Unicode Input” option under the Edit menu and browse through the on-line Unicode tables to check on the order of any characters.

Wild Cards Which Match Single Words

- * The asterisk character will match any sequence of characters in a word and therefore is the most versatile of the one word bounded wild cards. For example: “*es” will match “noses,” “proxies,” “heroines,” etc. It may be used at the beginning, middle or end of the response to match any character or group of characters except the space character.

Wild Cards Which Match Anything

This set of wild cards will function over a string of one or more words.

& The ampersand wild card will match anything, including the space character (). For example: “The chickens &” will match “The chickens are escaping,” “The chickens have demolished New York,” etc.

| (The Vertical Bar) The vertical bar wild card character | (usually above the backslash \ on most keyboards) separates explicit alternatives, any of which will match. The bar functions as an “or” operator. Thus “never|hardly ever” will match “never” and will also match “hardly ever” (but not “never ever”).

{ | } {response1|response2|response3|etc}
In cases where the vertical bar used alone would be ambiguous, braces are used in conjunction with the vertical bar to show how a series of anticipated responses are grouped. The “|” character functions as an “or” operator. For example: “The Baltimore Orioles are the {best|most feared|world champion} team” will match “The Baltimore Orioles are the best team”, “The Baltimore Orioles are the most feared team”, and “The Baltimore Orioles are the world champion team”.

Note: Braces may be used within braces, a technique called nesting, to provide even greater flexibility in anticipating responses.

Using WinCALIS Wild Cards in Anticipated Responses

Wild cards can be used in Correct/Continue answer statements, allowing several Correct/Continue alternatives to be included in a single answer statement. Consider the statement:

The {synonym|antonym|{noun|verb} clause} is correct.

This will match any of the following answers:

The synonym is correct.

The antonym is correct.

The noun clause is correct.

The verb clause is correct.

Note: Keep in mind that the student may use the “Give Up” feature to see expected response displayed. Therefore, this first Correct/Continue anticipated response should have no wild card characters, serving as a “model” correct answer.

Wild cards are used more often in anticipated Incorrect/Remain Here statements. They enable the author to check for whole classes of errors with a single statement. Consider the following example of anticipated Correct/Continue, Incorrect/Remain Here, and neutral answers together with respective feedback to the student:

Presentation (Text) Window:

Type the right form of the verb shown in parentheses.

Interactive (Question) Window:

The girl _____ the old woman daily. (to help)

Correct/Continue Anticipated Response (Short Answer format task):

helps

Feedback:

Yes, "The girl helps the old woman daily" is correct.

Anticipated Incorrect/Remain Here Answers:

1. *{help|help{ing|ed}}*

Feedback:

This is a wrong form of the verb. Try another ending.

2. **s*

Feedback:

The "-s" ending is correct, but use a form of "to help."

3. ***

Feedback:

This is incorrect. Please check your typing. If you are still having problems, review the example given in the directions (press [F9]) or review [Chapter ?, section ??].

4. *&*

Feedback:

We are looking for a one-word answer here.

Note: In the above examples the most specific answers come first and the most general last. Also note how the anticipated Incorrect/Remain Here response *** will catch any one-word answer, while the *'&'* wild card will catch any multi-word answer and reply to the student in an appropriate manner.

Below you can see the effect that Incorrect/Remain Here student answers will have as they are evaluated by the error anticipation and wild card system. By means of the wild cards, you can tell WinCALIS to look at a specific part of the text the student types and give an appropriate hint. This example has anticipated:

1) wrong ending

2) wrong stem

3) wrong word

Observe the relationship between the answers and the feedback.

Interactive (Question) Window:

The girl _____ the old woman daily. (to help)

Anticipated Student Responses:

1. *does help*

Feedback message:

We are looking for a one-word answer here.

2. *phones*

Feedback message:

The "-s" ending is correct, but use a form of "to help."

3. *helping*

Feedback message:

This is a wrong form of the verb. Try another ending.

4. helps

Feedback message:

Yes. "The girl helps the old woman daily" is correct.

Going Further with Wild Cards

To get an idea of how powerful your lessons can become by inserting wild cards, look at the following example of a script:

Interactive (Question) Window:

Where is the government of the U.S.A. located?

Correct/Continue Anticipated Response (Short Answer format tasks):

{DC|D.C.|Washington{,D.C.|,DC|DC|D.C.|} {[Tt]he| } Dist{,|ri|ct} of Columbia |Dist {,|ri|ct} of Columbia};

Feedback:

That's right!

Anticipated Incorrect/Remain Here Answer (Short Answer format tasks):

1. *{{d|D}}{c|C}}*

Feedback:

Check your spelling: <SPMK:DC>

2. *{{d.|D.|D|d}}{c|C|C.|c.}}*

Feedback:

Check your spelling: <CE><SPMK:D.C.>

3. *{{W|w}*n}*

Feedback:

Check your spelling: <SPMK:Washington>

4. *{{W|w}*n {D|d}{C|c}}*

Feedback:

Check your spelling: <SPMK:Washington DC>

5. *{{W|w}*n, {D|d}{C|c}}*

Feedback:

Check your spelling: <SPMK:Washington, DC>

6. *{{w|W}*{n,|n} {d.|D.|D|d}{c|C|C.|c.}}*

Feedback:

Check your spelling:<SPMK:Washington, D.C.>

7. *{the {D|d}is{t.|t} of {C|c}*}*

Feedback:

Check your spelling: <SPMK:the Dist. of Columbia>

8. *{the {D|d}istrict of {C|c}*}*

Feedback:

Check your spelling: <SPMK:the District of Columbia>

9. *{The {D|d}ist. of {C|c}*}*

Feedback:

Check your spelling: <SPMK:The Dist. of Columbia>

10. *{The {D|d}istrict of {C|c}*}*

Feedback:

Check your spelling: <SPMK:The District of Columbia>

11. *{{D|d}ist. of C*|dist. of C*|Dist. of c*|dist. of c*}*

Feedback:

Check your spelling: <SPMK:Dist. of Columbia>

12. *{District of {C|c}*}*

Feedback:

Check your spelling: <SPMK:District of Columbia>

13. *New York*

Feedback:

New York is an important American City, famous for its skyscrapers, but it's not the capital of the U.S.

14. *{{n|N}ew {Y|y}ork|{N|n}* {Y|y}*k}*

Feedback:

New York is an important American City, famous for its skyscrapers, but it's not the capital of the U.S.

By the way, check your spelling: <SPMK:New York>

15. *Los Angeles*

Feedback:

Los Angeles is big American City on the West Coast, famous for its film industry, but it's not the capital of the U.S.

16. *{{L|l}os {A|a}ngeles|{l|L}os {A|a}n*s}*;

Feedback:

Los Angeles is a big American City on the West Coast, famous for its film industry, but it's not the capital of the U.S. By the way, check your spelling: <SPMK:Los Angeles>

17. *San Francisco*

Feedback:

San Francisco is a West Coast city, famous for its Golden Gate Bridge and earthquakes, but it's not the capital of the U.S.

18. *{{S|s}an {F|f}rancisco|{S|s}*n {F|f}ra*o}*;

Feedback:

San Francisco is a West Coast city, famous for its Golden Gate Bridge and earthquakes, but it's not the capital of the U.S. By the way, check your spelling: <SPMK:San Francisco>

19. *San &*

Feedback:

Incorrect, the capital of the USA is an East Coast city.

20. *&*

Feedback:

No, <LAST> is not the capital of the US. Here's a clue: you might recognize this famous building found in the capital of the US. <GRAPHIC: WHITEHS>

In the above example, the Correct/Continue Anticipated Response (wild card) definition will consider the following answers as correct:

DC
D.C.
Dist. of Columbia
District of Columbia
the Dist. of Columbia
the District of Columbia
The Dist. of Columbia
The District of Columbia
Washington
Washington, DC
Washington, D.C.

If a student makes a spelling error in one of these possible entries, he will get specific spelling feedback because of the SPELLMARK command (<SPMK:[word]>) that was added to Anticipated Incorrect/Remain Here Answers (Anticipated Incorrect/Remain Here Answers 1 - 12). For a detailed discussion of the use of the SPELLMARK command in WinCALIS, refer to The Power of SPELLMARKing.

Anticipated Incorrect/Remain Here Answers (13 - 19) will cover the following student responses, and possible spelling mistakes against them:

New York
Los Angeles
San Francisco

The last Anticipated Incorrect/Remain Here Answer will catch any other student entry and give a clue to the student. The last student entry is printed to the Diagnostic Window through the inclusion of the <LAST> command in the feedback message. For a full discussion of the use of the <LAST> command in WinCALIS, see the section Displaying the LAST Student Entry and the CORRECT Answer. By means of wild cards, WinCALIS is able to handle an indefinite number of responses and provide the user with a matching feedback.

Spellmarking and Wild Card Use

By default, WinCALIS enables the SPELLMARK feature, which compares student responses to the Correct/Continue responses anticipated by the author. This feature has two important steps: first, it makes a comparison with all of the Correct/Continue responses to see which one it most closely matches. If it is close to one, Spellmark will return the 'marked up' response to the user, comparing it directly to that closest anticipated response. An anticipated response is not used for comparison in this way, however, if it has a wild card character in it.

Sequencing Answers and Wild Card Use

WinCALIS compares the student's answer with the list of Correct/Continue answers beginning with the first one listed for the task. If no match is found, WinCALIS compares the student response to the list of Incorrect/Remain Here answers in the same way. This has some important implications for the sequence of anticipated answers.

Correct Answer Sequencing

The first Correct/Continue answer listed in a task is the one shown to the student, if he or she gives up or reaches the maximum number of attempts allowed, when the author has checked the "Show Correct"

option in the Script Options or Task Options dialog box of WinCALIS Author. Any wild cards included in the first Correct/Continue answer will be shown, which may be confusing to the student.

Therefore it is important that the first Correct/Continue answer in any task be free from any wild cards; and be the most desirable student answer possible.

Anticipated Incorrect/Remain Here Answer Sequencing

When anticipating student errors, an author may want to use wild cards to anticipate a specific error, and also a more general form of the same class of error. Consider these two WinCALIS scripts, which may, at first glance, appear quite similar.

Script I

Question:

She is _____ to John. (to talk)

Correct/Continue Anticipated Response (Short Answer format tasks):

talking

Feedback:

Yes, "She is talking to John." is correct.

Anticipated Incorrect/Remain Here Answers (Short Answer format tasks):

1. *talks*

Feedback:

"She talks" could be right, but here you need a different form.

2. *talk**

Feedback:

This sentence requires a present participle.

Note: In the Anticipated Incorrect/Remain Here Answers sequence, the last anticipated Incorrect/Remain Here response (*talk**) is the most general, that is, it will match "talking," "talks," "talked," an addition to meaningless responses such as "talkx," etc.

Script II

Question:

She is _____ to John. (to talk)

Correct/Continue Anticipated Response (Short Answer format tasks):

talking

Feedback:

Yes, "She is talking to John." is correct.

Anticipated Incorrect/Remain Here Answers (Short Answer format tasks):

1. *talk**

Feedback:

This sentence requires a present participle.

2. *talks*

Feedback:

"She talks" would be right, but "She is talks" is not correct. Please try a different form of the verb.

In this sequence of Anticipated Incorrect/Remain Here Answers, the last response (talks) is not the most general. It will only match "talks" as a student response.

In the first script (see Script I) the specific anticipated Incorrect/Remain Here response *talks* comes before the general *talk**. In the second (see Script II) it follows. WinCALIS evaluates these in order; the first matching anticipated Incorrect/Remain Here response is the one the student will receive. Because the student response *talks* matches both first and second anticipated Incorrect/Remain Here response in both scripts, the more specific feedback will never appear. This leads to the general rule:

ALWAYS put more specific anticipated Incorrect/Remain Here responses ahead of more general ones.

Another question of order in anticipated Incorrect/Remain Here responses arises when the author anticipates more than one type of error for a given question. For instance, in the following script the author has anticipated a problem with the plural form, an error in the placement of the adjective, and an incorrect form of the verb. Here is an example of anticipating four different errors:

Interactive (Question) Window:

The red church is large. (Make the sentence plural)

Correct/Continue Anticipated Response (Short Answer format tasks):

The red churches are large.

Anticipated Incorrect/Remain Here Answers (Short Answer format tasks):

1. *& churchs &*

Feedback:

The plural of church is formed with "-es," not "-s."

2. *& is &*

Feedback:

'Is' is a singular form. A plural form of the verb "to be" is required.

3. *& {reds|larges} &*

Feedback:

Do not put an "-s" at the end of an adjective. Adjectives are not modified to form plurals.

4. *&*

Feedback:

The noun and the verb change form here. The other words look the same.

In order to anticipate grammatically different errors, they must be anticipated one at a time. It is conceivable that a student's answer would make all three mistakes anticipated in the above script (e.g. "The reds churchs is large"). In this instance, WinCALIS will not point out all three errors at once. Rather it will help the student with the special plural form first. When the student has the plural correct, adjective placement will be tested. After everything else is correct, the student will be warned about the verbal form. The sequence is controlled by the order of anticipated Incorrect/Remain Here responses in the example script. Thus the following rule should be kept in mind:

When more than one type of error is anticipated, arrange anticipated Incorrect/Remain Here responses so that the mistakes are addressed in order of importance: most important comes first.

The WinCALIS feedback and wild card system is open-ended. On one hand, it is possible to check various student responses by using many different Anticipated Incorrect/Remain Here Answers statements; and on the other, it is possible to check many responses with just a few lines through use of wild cards. You will find that experience in authoring as well as testing scripts with real students are important factors in creating comprehensive answer analysis. The way WinCALIS handles Answer Sequencing at the level of the WinCALIS Authoring Language will be explained further in the section CALIScript.

Using Wild Card Characters as Literal Characters

By their very nature, wild card characters never stand for themselves; they always signify something different. Thus some adaptation must be made when you want one of the wild cards to appear within an actual student response, such as “Jack & Jill.” In order to deactivate the wild card function of a particular wild card character, you must precede it with a backslash, \. Thus the anticipated response, “Jack \& Jill” will match only “Jack & Jill” as student input, while the anticipated response “Jack & Jill” will match “Jack loves Jill” “Jack pushed Jill,” etc. In addition to the wild cards mentioned so far, the backslash must be used to anticipate literal user input of the semicolon (;) character and the backslash character itself, e.g. “Directories are always preceded with a \\ character.”)

Note: Wild card characters used literally in anticipated responses, preceded by a backslash (e.g., *) usually must be preceded by two backslashes in WinCALIS 2.x (e.g., *).

Using Special Responses in WinCALIS Author

In most cases students complete questions or tasks in a WinCALIS lesson by typing in a character or set of characters and pressing ENTER. Just pressing ENTER takes the student to the next task. However, other options called Special Responses do exist. As stated in the section Getting Started with WinCALIS, unless otherwise specified, students are allowed to pass among tasks either by pressing the ENTER key by itself or by pressing one of the gray “plus” or “minus” keys on the numeric keypad. A student also may press the F6 function key and see the Correct/Continue answer if at least one attempt has been made to complete the task.

At times it may be convenient for an author to be able to anticipate these responses and link appropriate feedback to them. This is possible by entering the appropriate commands in the Anticipated Response box of the Short Answer dialog box in WinCALIS Author. The array of choices may be browsed or inserted using the ‘Insert Special Response or Wild Card’ drop-down list box in the Short Answer dialog box.

A single WinCALIS task may contain different Special Responses in addition to the text strings which make up anticipated Correct/Continue and Incorrect/Remain Here student responses. Special Responses are always evaluated by WinCALIS before other anticipated responses. There are a number of Special Responses that can be entered in the Anticipated Response box of the Short Answer dialog box: <ENTER>, <GIVEUP>, <PASS>, <PASSBACK>, <KEY 'x'>, <MOUSE>, <BUTTON>, <HOTSPOT>, and <IF x>. In addition, the Special Response <OR> is used in combination with two or more other Special Responses.

Using Special Responses: <BUTTON>

See Pressing a Button.

Using Special Responses: <ENTER>

The <ENTER> special response is probably the simplest one, although all special responses function similarly. The special response is entered in the Anticipated Response box of the Short Answer dialog box. The author can, if necessary, attach appropriate feedback to it by entering this in the Feedback

box. <ENTER> can be used as a Correct/Continue, Incorrect/Remain Here, or neutral response. For example:

Presentation (Text) Window:

This concludes the first section of this lesson. Press ENTER to continue to the next session.

Correct/Continue Anticipated Response (Short Answer format tasks):

<ENTER>

Feedback:

You will now move to the second section of the lesson.

From the example above you may already guessed that <ENTER> as a special response is especially useful when the author merely wants to present information to the student, instead of asking a question. By pressing the ENTER key during the learning session, the student completes the task and moves to the next.

Another example of functional use of pressing the ENTER key as a response could be to provide a hint to the student. You can find a more detailed discussion of this in the section CALIScript, under "The Enter Key."

Using Special Responses: <GIVEUP>

By using the <GIVEUP> special response you can tell a student what to do upon giving up. The procedure to use this special response is the same as above: simply enter it in the Anticipated Response box of the Short Answer dialog box.

Interactive (Question) Window:

What city is the capital of Albania?

Correct/Continue Anticipated Response (Short Answer format tasks):

Tirana

Feedback:

Correct

Anticipated Incorrect/Remain Here Answer:

<GIVEUP>

Feedback:

Tirana is the capital of Albania, which is a country in Southern Europe.

If a student gives up by pressing the F6 key, the Correct/Continue answer is shown unless the Show Correct option in the WinCALIS Author Script Options dialog box is deselected. Specifying the <GIVEUP> response as an anticipated response allows a feedback message other than the first Correct/Continue answer to be displayed. In this case the message shown is, "Tirana is the capital of Albania, which is a country in Southern Europe." The <GIVEUP> special response will not work if the <NOGIVEUP> Task Option is in effect. You can learn more about Task Options in the section CALIScript. Most of the WinCALIS options are available in the Script Options or Task Options dialog boxes in WinCALIS Author.

It is also possible to require more than one attempt before allowing a student to use the GIVEUP key by specifying the minimum number of tries before the student can give up from the Script Options dialog box.

Using Special Responses: <HOTSPOT>

<HOTSPOT name [0,0,100,100]>

Defines action by WinCALIS if student clicks on a hot area of a graphic. By default 100%, restricted to a smaller area by parameters.

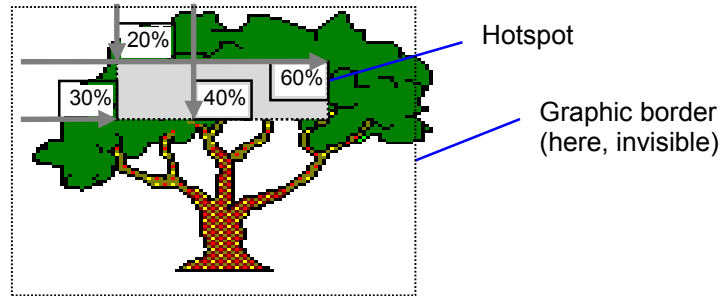


Figure 14: Indicating a hotspot on a graphic <HOTSPOT 20, 30, 40, 60>

See Making Hotspots in Graphics

Using Special Responses: <IF>

<IF:response\$=variable\$>

<IF:mark\$=variable\$>

<IF:numvariable=value>

Defines action by WinCALIS if the student's text response, stored in response\$, or text marked, stored in mark\$, matches the value of variable\$, or expression numvariable=value returns true.

See Also: Acting on a Special Response IF a Condition is Satisfied
Variables

Using Special Responses: <KEY 'x'>

Usually a student must press the ENTER key after typing in an answer, in order to send the response to WinCALIS to be evaluated or acted upon. With the special response <KEY: 'x'> it is possible for the author to designate a keystroke which is interpreted by WinCALIS, immediately without being followed by ENTER.

Interactive (Question) Window:
Press "b" to begin.

Correct/Continue Anticipated Response (Short Answer format tasks):

<KEY 'b'>

As soon as the student presses the “b” key, the current task is completed and the user can move to the following tasks in the script. Combination keystrokes may also be defined with this special response using the Shift and Ctrl keys. All of the following are valid:

```
<KEY:SHIFT 'b'>  
<KEY:CONTROL '5'>  
<KEY:CONTROL SHIFT 'x'>
```

Remember that WinCALIS special responses are evaluated before other anticipated student responses. Be careful not to combine <KEY: 'x'> special responses with anticipated text answers that begin with the same letter. For example:

Anticipated Incorrect/Remain Here Answers (Short Answer format tasks):

1. sure
2. <KEY:'s'>

As soon as the student presses the “s” key, WinCALIS matches the input with the <KEY:'s'> special response. It is impossible for the student to type “sure” or any word beginning with an “s.”

There are also the following extensions of the <KEY> command:

<KEY ANY>

This will take effect if any key at all is pressed.

<KEY controlkey>

This variation enables you to specify virtually any of the “control” keys on the keyboard. Substitute one of the keywords in the list below for the word 'controlkey' in the command.

SHIFT, CONTROL, ALT, BACK, TAB, CAPITAL, ESCAPE, SPACE, PRIOR, NEXT, END, HOME, LEFT, UP, RIGHT, DOWN, DELETE, NUMPAD0, NUMPAD1, NUMPAD2, NUMPAD3, NUMPAD4, NUMPAD5, NUMPAD6, NUMPAD7, NUMPAD8, NUMPAD9, MULTIPLY, ADD, SEPARATOR, SUBTRACT, DECIMAL, DIVIDE, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, NUMLOCK

Using Special Responses: <MOUSE: x y>

With WinCALIS it is possible to define a region of the screen and allow the student to complete a task by using the mouse to point and click on that region. This is done using the <MARK:x> command and the special response <MOUSE: x y>. You can learn more about the use of the <MARK:x> command later in this section (Positioning the Cursor at Specific Marks in Your Text or Question) and in the <MARK:x> section of CALIScript.

Syntax: <MOUSE: x y>

Parameters: “x” and “y” should refer to the upper left-hand and lower right-hand corners of the region in which a student may point and click to give an answer. These corners are defined by means of the <MARK:x> command.

Popup (Directions) Window:

Point and click on the city which is the US capital.

Interactive (Question) Window:

<MARK:p1>Paris<MARK:p2>

<M:w1>Washington<M:w2>
<M:l1>Lima<M:l2>

Correct/Continue Anticipated Response (Short Answer format tasks):
<MOUSE w1 w2>

Feedback:
Good! Washington is indeed the US capital.

Anticipated Incorrect/Remain Here Answers

1. <MOUSE p1 p2>
Feedback:
Paris is the capital of France.

2. <MOUSE l1 l2>
Feedback:
Lima is the capital of Peru.

In the above example, the <MARK:x> command is used to define the area in the Interactive (Question) window where the student can point and click using the mouse to answer a question. As you can see the <MARK:x> command can be abbreviated to <M:x>. The two “marks” defined with the <MARK:x> command define the upper left-hand and lower right-hand corners of the region in which a student may point and click to give an answer. This area may be a single character or line or a much larger area. See the section on <MARK> for more details.

When the student moves the cursor over an area which has been marked in the manner described above the cursor changes shape from an arrow to a pointing hand, which means that the area can be clicked on and WinCALIS will respond.

The special response <MOUSE: x y> is then used to define how these answers will be judged and what action will follow them. The “x” and “y” parameters should be filled with the corresponding parameters from the <MARK:x>.

For marking “hot” areas on graphics, the <MOUSE: x y> special response has been largely supplanted by the <HOTSPOT> special response.

For more examples, see “<MARK>ing for Hot Text”.

Using Special Responses: <OR>

The <OR> special response operator is a special kind of special response. It was added in order to make authoring even easier. In some instances you may want WinCALIS to respond exactly the same way to two or more events, such as the click of the mouse on a certain hot link *or* the press of a certain key. You can use the <OR> special response to anticipate these two events in a single answer by linking two special responses with <OR>.

Interactive (Question) Window:
Click <M:s1>Start<M:s2> or press 's' to begin this session.

Correct/Continue Anticipated Response (Short Answer format tasks):

<MOUSE:s1 s2><OR><KEY 's'>

Both clicking the word “Start” in the Interactive (Question) Window or pressing “s” on the keyboard will cause WinCALIS to continue to the next task, since the responses are marked as correct in the Short Answer dialog box.

The <OR> special response can be used to link more than two special responses, but it may not be used to link text responses with special responses.

Using Special Responses: <PASS>

The <PASS> special response allows the WinCALIS author to anticipate a student passing over a task without completing it and execute a set of commands in response to this action. Unless otherwise specified, students are allowed to pass among tasks by pressing the gray plus key on the numeric keypad or the Ctrl+PageDown cursor control key. Passing over one task will bring up the next one.

This option can be restricted using the Allow Passing task option, which you can deselect in the WinCALIS Author Task Options dialog box. You also may prohibit students from passing among tasks in an entire WinCALIS script by deselecting the Allow Passing option in the WinCALIS Author Script Options dialog box. You can learn more about the <NOPASS> command in the section CALIScript in the section on Task Options. In some cases, the author may wish to allow the student to pass between tasks but the design of a lesson requires that some action be taken when a student passes a task without completing it.

Consider the following example, in which <PASS> is used in order to rewrite the Interactive (Question) Window if student passes:

Interactive (Question) Window:

<BO>___ am happy.<BO>

What pronoun completes the above sentence?

Correct/Continue Anticipated Response (Short Answer format tasks):

I

Feedback:

Very good. This is correct.

<QUESTION><BLANK> I am happy.

Anticipated Incorrect/Remain Here Answer (Short Answer format tasks):

<PASS>

Feedback:

<QUESTION><BLANK> ___ am happy.

In the above example, the author has decided to rewrite the sentence in the Interactive (Question) Window after the student's response; the correct student answer is entered in the gap of the sentence in the Interactive (Question) Window. It appears originally with the gapped sentence bolded. The <BOLD> or abbreviated <BO> command will be discussed in the section The Bold, The Strikeout, and Others in this section. If the student presses the Pass key instead of answering the question, he will move on to the next task. The presence of the <PASS> command as an Anticipated Incorrect/Remain Here Answer triggers a specific action by WinCALIS. In this case, the gapped sentence in the Interactive (Question) Window will be rewritten so that it is no longer bolded. The <QUESTION> command is one of the Window Activation commands that you can learn further about in this section (see Addressing Windows when Giving Feedback). The <BLANK> command will be explained in the section Blanking a Window of CALIScript.

The <PASS> response should be used only in Anticipated Incorrect/Remain Here or neutral Answers.

Using Special Responses: <PASSBACK>

The <PASSBACK> special response is very similar to the <PASS> special response. In fact, the <PASSBACK> special response is only different from the <PASS> special response in one very important way. It matches the student response created by pressing the gray minus key or Ctrl+PageUp cursor control key. It does not match the student response ENTER key, the gray + key, or the Ctrl+PageDown cursor control key. If there is no difference between passing forward among tasks in a lesson and passing backwards, then the <PASS> special response is all that is required to allow the WinCALIS author to define what commands should be executed if a student passes a task. If it is necessary to make a distinction between passing forward and passing backwards among tasks, then the <PASSBACK> special response is used. In this case the effect of the <PASS> special response is narrowed to match only the ENTER key, the gray + key, or the Ctrl+PageDown cursor control key.

Aesthetics of Lesson Presentation

After you decide upon just the right text, questions, and feedback, you have one major task left--packaging it. To fully take advantage of the interactive nature of computer-assisted language learning, you should make the lesson inviting and stimulating in a visual way. With this in mind, WinCALIS Author gives you several options to enhance your lesson. Video and graphics constitute the most striking visual effects, but you will also want to know about selection for your windows and their associated text.

Color Me CALIS

You can really spice up your text by employing the Format menu in WinCALIS Author. Text in any window may be displayed in any of twenty-two colors: black, blue, dark blue, brown, cyan, dark cyan, dark gray, light gray, green, dark green, light green, olive, orange, pink, peach, purple, dark purple, red, dark red, white, yellow, and light yellow. Colors can be mixed within any given window to add emphasis or extra meaning to a group of words, or you can put the text in each window in different colors. You could put directions in green, text in blue, questions in red, or whatever color scheme you like.

In WinCALIS Author, color schemes are controlled by menu selections. Just mark the portion of the text you want to color by holding down the left mouse button and dragging the mouse pointer on the word(s) to be marked. Then select the Format menu and the Mark: Color submenu. You can select any one of the twenty-two colors, and WinCALIS Author will automatically insert color commands like <BLUE> or <GREEN> surrounding the word(s). The first command turns the color on and the other turns it off. In other words, the first appearance of the <color> command turns it on until the same color command is inserted again to turn that color off. For example, if the statement:

this is the color <RED>red<RED>

was inserted in the Presentation (Text) Window after you had selected the color RED from the Format, Mark: Color menu, then the word “red” would appear in red typeface when run in WinCALIS.

You can display all the available colors using a “dummy” lesson with the color commands in one of the windows. You might type the names of the available colors in the Presentation (Text) Window, then mark them and have the color commands automatically inserted by selecting the appropriate color options from the Format: Mark: Color menu. (You can also copy this topic from the copy of the User's Manual in on-line Help, then paste the commands below right into your Author window.) For example:

Presentation (Text) Window:

These are the 22 colors that WinCALIS displays:

```
<BLUE>Blue<BLUE>
<BLACK>Black<BLACK>
<YELLOW>Yellow<YELLOW>
```

```
<WHITE>White<WHITE>
<GREEN>Green<GREEN>
<CYAN>Cyan<CYAN>
<PURPLE>Purple<PURPLE>
<OLIVE>Olive<OLIVE>
<PINK>Pink<PINK>
<PEACH>Peach<PEACH>
<BROWN>Brown<BROWN>
<ORANGE>Orange<ORANGE>
<LTGRAY>Lt Gray<LTGRAY>
<RED>Red<RED>
<LTGREEN>Lt Green<LTGREEN>
<LYELLOW>Lt Yellow<LYELLOW>
<DKGRAY>Dk Gray<DKGRAY>
<DKBLUE>Dk Blue<DKBLUE>
<DKRED>Dk Red<DKRED>
<DKGREEN>Dk Green<DKGREEN>
<DKPURPLE>Dk Purple<DKPURPLE>
<DKCYAN>Dk Cyan<DKCYAN>
```

You must remember that WinCALIS always needs two identical color commands (one to turn color on; the other to turn color off) before attempting to display text in a new color. The end of a line will also serve as the 'color off' switch. (For applying color to more than one line, see Setting Foreground and Background Colors)

Here's another example of color formatting:

What would happen if <BLUE>Little Boy Blue<BLUE>
got together with <RED>Little Red Riding Hood<RED>?

Would it be the start of a Rainbow Coalition or
would their children be <PURPLE>Purple People Eaters<PURPLE>?

As you can see, color can give your lesson a style that will keep your students visually stimulated, and thus more receptive for learning.

Setting Foreground and Background Colors

If you want all the text in a particular window to be of one color, or if you want the background to be of some solid color, you may use the Foreground Color and the Background Color options from the Format menu in WinCALIS Author. By inserting the <SET [FCOLOR|BCOLOR] Color> command in your text, you specify the window's total appearance for as long as it's on the screen or until you make changes yourself by having a new <SET [FCOLOR|BCOLOR] Color> command inserted.

The default foreground color is black; the default background color is white.

The Set: Foreground Color menu option allows you to choose from twenty-two different colors. Selecting one of the colors from this menu will prompt WinCALIS Author to insert the chosen <SET FCOLOR Color> command, and when the lesson is run in WinCALIS, to display all of the text in the chosen color. The Set: Background Color option allows for the same colors. The Background Color option will insert commands in your text that affect the background color of your windows. Obviously, you can also insert <SET [FCOLOR|BCOLOR] COLOR> in the windows yourself. In that case you should be careful to use the right syntax.

The full syntax for these commands is as follows:

Foreground:

```
<SET: [W] FCOLOR Color>
```

Background:

<SET: [W] BCOLOR Color>

Window:	Syntax:	Abbreviation
Popup (Directions)	<DIR>	none
Interactive (Question)	<QUESTION>	<Q>
Presentation (Text)	<TEXT>	<T>
Tutorial	<F[1-10]>	none
Diagnostic Feedback	<D>	none

Note: If you wish to set the foreground or background colors of the Lesson Description Window, you must use the <SET: color> command within the window itself. The Lesson Description Window can never be referred to in a command in another window. (Nor would this be very useful, since the Lesson Description Window has already come and gone by the time any other window is created.) The Diagnostic Feedback Window is also a special case. The D Window is created anew in every task and destroyed when the task is completed. So any command which references the D Window will have an effect only if the D Window currently exists and only for the current task. In order to set the colors of the D Window reliably, the <SET: color> command should appear in each D window for which the colors are desired. One way of accomplishing this semi-automatically is to insert <SET: color> commands in the Script Options “Default correct feedback:” and “Default incorrect feedback:” boxes.

Color Options:

WHITE	LTGRAY	DKGRAY
PURPLE	DKPURPLE	BLUE
DKBLUE	CYAN	DKCYAN
OLIVE	LIGHT GREEN	GREEN
DKGREEN	LIGHT YELLOW	YELLOW
RED	DKRED	PINK
PEACH	BROWN	BLACK

It should be noted that if you do not specify any window in the command, the color will affect the active window, the one you are currently working in. For example, if you insert a <SET [FC|BC] color> command in the Presentation (Text) Window, it will affect the Presentation (Text) Window. For more information about Window activation commands in WinCALIS, refer to the section Addressing Windows when Giving Feedback--The Window Commands and the section CALIScript.

Note: The WinCALIS opening screen (WinCALIS Author Workstation only) allows the author or system administrator to select color and typographic attributes for each window in the Window Defaults... menu item under the Options menu.

The foreground color command sets the default color in the specified window ([W]) to color. <color> commands still function as usual, but when a new color is deactivated with a second <color> command, the type will return to the default color specified in the <SET: [W] FCOLOR Color> command.

The <SET: [W] FCOLOR Color> command is active until it is overridden by another <SET: [W] FCOLOR Color> command. This means it might be used to color a segment as small as one character up to a full script. The <SET: [W] BCOLOR Color> Color should only be set once within a task since any changes within a task are instantaneous and only the final effect of the <SET: [W] BCOLOR Color> appears on screen. <SET: [W] BCOLOR Color> and <SET: [W] FCOLOR Color> commands override the Default Color Options as set in the Windows Defaults menu on the WinCALIS opening screen. The following are all valid Color commands:

```
<SET: TEXT FCOLOR GREEN>
<SET: TEXT FC GREEN>
<SET: T FC GREEN>
<SET: QUESTION BCOLOR BLUE>
<SET: Q BCOLOR BLUE>
```

<SET: Q BC BLUE>

Taken together, the above commands cause WinCALIS to appear in green type on a blue background. But we can change the type to <RED>red<RED> before it is switched back to green.

Note: You may abbreviate “FCOLOR” as “FC” and “BCOLOR” as “BC.” Also remember that if you do not specify a Window, the color change will affect the current window.

A Last Note on Color Use

Although colors might brighten up the look of your lesson, you should be aware that too much color or a wrong match of foreground and background colors might be confusing and difficult to read. Remember that dark text on a light background in a combination of contrasting colors is generally considered by experts to be the best match. Why not use colors in a functional way? **Red** is generally associated with the concept “Incorrect,” **green** or **blue** with the concept “Correct.”

The Bold, The Strikeout, and the Others

Using the same method as coloration, you can make your text appear in certain attributes like bold, reverse video, underline, and strikeout. (Reverse video switches the background and foreground colors for the chosen text.) These attributes are generally used to highlight specific portions of text and can be used in conjunction with the coloration commands any time you display text in the lesson. Highlight (“mark”) the text you want to apply the attribute to, then select the attribute you wish in the Mark: section of the Format pull-down menu. WinCALIS will insert the correct attribute commands surrounding the text you highlighted. The first appearance of the attribute command will turn the attribute on, the second will turn it off. The attributes are listed with their commands in angled brackets below.

<BOLD>Produces bold text<BOLD>
<ITALIC>Produces italic text<ITALIC>
<REVERSE>Produces reverse video text<REVERSE>
<STRIKEOUT>Produces strikeout text<STRIKEOUT>
<UNDERLINE>Produces underlined text<UNDERLINE>

Each of these commands has a convenient abbreviation:

<BOLD>	<BO>
<ITALIC>	<IT>
<REVERSE>	<RV>
<STRIKEOUT>	<SO>
<UNDERLINE>	

Contrary to the coloration commands, you can make combinations of attribute commands. For example:

<BO><IT>Bold and Italic<BO><IT>

will cause the text between the attribute commands to be displayed in bold and italic. Attribute commands might also be combined with coloration commands such as in these examples:

<RED><RV><IT> Red, Reverse, and Italic <IT><RV><RED>

will display white text in italics on a red background, for just the text marked.

<BLUE><BO> Blue and Bold <BO><BLUE>

will display blue text in bold.

<SET FC DKGRAY><RV><BO>Dark Gray, Reverse and Bold<BO><RV><SET FC BLACK>

will display white text in bold on a dark gray background.

<SET FC DKGREEN><SO>Strikeout<SO> and Underline<SET FC BLACK>

will produce green text partly in strikeout, partly underlined.

Positioning the Cursor and Text in Windows

Perhaps one of the most fundamental concepts for WinCALIS screen design is that of the cursor location. Most word processors employ a flashing cursor that invites users to type a character. The location of the flashing cursor indicates where the next letter will appear on the screen. As a user types more characters, the cursor moves to the right and down the screen.

WinCALIS also uses a cursor, but only in one window (the Interactive (Question) Window), and only when the student user is being asked to type an answer. Every window, however, has a cursor location, i.e. the row and column within the window where the next character will appear. The location also moves to the right and down as text is added to the window. It might be useful to consider that every WinCALIS window maintains a cursor, which differs from the flashing cursor used by word processors only in that WinCALIS cursors are usually invisible.

WinCALIS allows authors to change the location of the cursor. The command `<CURSOR:#r,#c>` changes the location of the cursor and the location within a window where the next character will appear. `<CURSOR:#r,#c>` moves the cursor location to the row and column specified. The first number is the number of lines from the top of the window. The second number is the number of spaces from the left. The upper left cell of a window is an absolute position (1,1) and acts as a starting point.

Syntax: `<CURSOR:#r,#c>`

Abbreviation: `<CUR:#r,#c>`

Parameters: #r-the absolute row number in the window
 #c-the absolute column number in the window.

For example:

Interactive (Question) Window:

`<CUR:3,25>`Who is buried in Grant's tomb?

In the above example the cursor is moved to line 3, space 25 in the Interactive (Question) Window. The `<CURSOR:#r,#c>` command is used to redirect the cursor either in the window in which it is used or in another window. It is often used with commands such as `<MARK>`, `<TEXT>`, `<BLANK>`, and `<TOPLINE>`. These commands are explained in other sections of this section.

The `<CURSOR:#r,#c>` command can be used with a pair of absolute values to move the cursor to a point in a window as shown in the example above, but it can also be used with relative values. In such a case you would specify how many lines and columns you want to move the cursor left, right, down or up from the current position of the cursor.

Syntax: `<CURSOR:+#r,-#c>`

Abbreviation: `<CUR:+#r,-#c>`

Parameters: `+#r/-#r` --add or deduct the number of rows from the current cursor position.

`+#r/-#r` --add or deduct the number of columns from the current cursor position.

For example:

Interactive (Question) Window:

```
<FS LARGE><CUR:1><CUR:,40>top  
<CUR:+7,-42>left<CUR:;,+33> center <CUR:;,+35> right  
<CUR:+7,-45>
```

The <CURSOR> commands as used in the above example will cause WinCALIS to display the words top, left, center, right, and bottom in a cross-shaped form. The first command (<CUR:1>) positions the cursor on the first line, the second command (<CUR:;,40>) positions the cursor 40 spaces to the left, the third cursor command (<CUR:+7,-42>) moves the cursor 7 lines down and 42 columns to the right, etc. If no number is specified for rows or columns in a <CURSOR:;#r,#c> command, then WinCALIS will assume the same value as the current value. You should remember two basic rules:

- One, whenever you move to a next line when typing your text and commands in the window, WinCALIS considers this as a +1 row value. If the third cursor command (<CUR:+7,-42>) would have been on the first line in the window, you would have had to change it into (<CUR:+8,-42>) in order to reach the same position. Equally, each character you type is seen as 1 column value. Typing the word “left” would mean that the column value is increased by four, because the word consists of four characters.
- Two, you should make sure that the CENTER option(s) from the Script Options dialog box is deselected. If not it will conflict with the positions you specify. There is still a third way to position the cursor, in which you don't have to worry about screen coordinates. In order to do so, you should use the <MARK:x> command, which is explained further on in this section. For a detailed discussion of the use of the <CURSOR:x> command in interaction with the <MARK:x> command, refer to the following topics in this section (See Positioning the Cursor at Specific Marks in Your Text or Question).

See Also: <CURSOR>

Sizing Up Your Text

Another typographic attribute that you may find handy is the ability to size portions of text. Using commands in angled brackets, you may select text to be sized as Tiny, Fine, Small, Medium, Large, Extra Large, and Huge.

Sizing works much like the color and typographic commands, in that you select type sizes using commands in angled brackets for your desired text selection. For example:

<FONTSIZE LARGE>Large

would display the word “Large” in a large font size. There are two slightly different commands for sizing. The first, <FONTSIZE: Size> selects a text size for the current line of text with the Size options TINY, FINE, SMALL, MEDIUM, LARGE, XLARGE, HUGE. You may abbreviate FONTSIZE to FS. The <FONTSIZE: Size> command only affects font sizing until the end of current line or until another font sizing command is encountered in the same line.

Syntax: <FONTSIZE: Size>

Abbreviation: <FS Size>

Size Options:

TINY • FINE • SMALL • MEDIUM • LARGE • XLARGE • HUGE

<FS FINE> will display font size fine text until the end of the current line or until <FONTSIZE MEDIUM> or another <FONTSIZE FINE> command is encountered on the same line.

The second variation of the command selects a size for the entire window in which it appears until another sizing command is encountered in that window. The syntax for the command is <SET: [W] FONTSIZE Size> with the same SIZE options and abbreviation [FS]. The Window parameter is optional when creating lessons in WinCALIS Author, but if used, should give the name of the window in which the text is to appear, i.e. TEXT, QUESTION, D or a tutorial window F[1-12]. When using this command in the feedback window, you must leave out the Window parameter.

Syntax: <SET: FONTSIZE [W] Size>

Abbreviation: <SET: FS [W] Size>

Window:	Syntax:	Abbreviation
Popup (Directions)	<DIR>	none
Interactive (Question)	<QUESTION>	<Q>
Presentation (Text)	<TEXT>	<T>
Tutorial	<F[1-10]>	none
Diagnostic Feedback	<D>	none

Note: If you wish to set the font size of the Lesson Description Window, you must use the <SET FONTSIZE fontsize> command within the window itself. The Lesson Description Window can never be referred to in a command in another window. (Nor would this be very useful, since the Lesson Description Window has already come and gone by the time any other window is created.) The Diagnostic Feedback Window is also a special case. The D Window is created anew in every task and destroyed when the task is completed. So any command which references the D Window will have an effect only if the D Window currently exists and only for the current task. In order to set the fontsize of the D Window reliably, the <SET FONTSIZE fontsize> command should appear in each D window for which the font size is desired. One way of accomplishing this semi-automatically is to insert <SET FONTSIZE fontsize> commands in the Script Options “Default correct feedback:” and “Default incorrect feedback:” boxes.

Size Options:

TINY • FINE • SMALL • MEDIUM • LARGE • XLARGE • HUGE

Because the <FONTSIZE: Size> command affects line sizing only until the next line break (or soft return character) in the window, it is suggested that you use the <SET: [W] FONTSIZE Size> command to switch between sizes as needed.

Note: The command <SET: FONTSIZE Size> can be automatically inserted in the window of your choice by selecting it from the Set Fontsize option in the Format menu in WinCALIS Author.

Here is an example:

Presentation (Text) Window:

<SET TEXT FS TINY> Little Bo Peep lost her sheep and couldn't figure out where they went. <SET TEXT FS FINE> She was in a fine mess because she had been taking care of her friend Mary's little lamb. <SET TEXT FS MEDIUM> “What might have happened,” she thought, and began to despair <SET TEXT FS LARGE> Crying in the meadow, she looked up and saw the Big Bad Wolf come upon her. <SET TEXT FS XLARGE> His teeth were glittering and Little Bo <SET TEXT FS HUGE> SCREAMED! <SET TEXT FS MEDIUM> The wolf then politely apologized. <SET TEXT FS SMALL> (He didn't like those sheep anyway.)

Interactive (Question) Window:

What did little Bo do when she saw the wolf coming closer?

The default font size for the Presentation (Text) Window is SMALL. When you change sizes, be sure to examine the lesson in WinCALIS to be sure that the text has been fitted correctly. In the above lesson, hard returns (pressing the return or ENTER key) had to be inserted to make the HUGE sized text fit in the window.

CENTERing a Text Line

From the Script Generation Options menu in WinCALIS Author, you can select Presentation (Text) Window from the Center Text in the Windows options group. This will cause WinCALIS to center all of your text in the Presentation (Text) Window. There may be times, however, when you just want to center one or a few lines. You can do so by entering the <CENTER> command in front of the line to be centered in any of the WinCALIS windows. This command only affects the line in which it appears, and is ideal to center a title, or to make a particular (set of) line(s) stand out. The <CENTER> command will work in all WinCALIS windows.

Syntax: <CENTER>

Abbreviation: <CE>

Presentation (Text) Window:

The New York Times, 17th April 1992, Fiona Galway
<CE><BO>The Night New York was Left in the Dark

In this example the second line, which contains the title to the text, is centered on screen. Combined with screen attributes such as color, font size, and appearance, the <CENTER> command may greatly enhance the look of a lesson.

See Also: Centering Text On-Screen

Using TABs to Align Text On-Screen

Another command that can improve the typography of your lessons is the <TAB:n> command. As easy as the <CENTER> command, it can be entered in any WinCALIS window and used with every Task format.

From the Script Generation Options dialog box in WinCALIS Author, you can select and modify the Tab Stop option listed under More Options in the Multiple Choice Task dialog box. This menu option, however, will only affect the position of multiple choice items in the Interactive (Question) Window when presented to the student. It will not affect the other Task Formats or other WinCALIS windows. If you would like to align text in the Presentation (Text) Window, for example, the <TAB:n> would make perfect tool to do so. Just insert the command in front of the text. The <TAB:n> command only works for the line in which it is inserted. The <TAB:n> command is equal to forty pixels (about eight spaces). The advantage over spaces is that the <TAB:n> command is not affected by the problems caused by proportionally spaced fonts when aligning text. For example:

Presentation (Text) Window:

1. <TAB:1>one
2. <TAB:1>two
3. <TAB:1>three
- ...
10. <TAB:1>ten

In the example the numbers spelled in words will be aligned to the left, even though the two digits of “10” take up more spaces than “1” or “2.”

Syntax: <TAB:n>

no abbreviation

Parameter: n—an absolute value; the highest value depends on the font size that is used.

If in the Presentation (Text) Window you write,

<TAB:1><TAB:1><TAB:1>text

the word “text” will appear at tab stop 1, although three <TAB:#n> commands were used. If you would like to position the word “text” at the tab stop 3, you should use this command instead:

<TAB:3>

Note: The number used with the <TAB:n> command is absolute. The <TAB:1> does not work cumulatively.

The <TAB:n> command is a handy tool to create neat looking columns in WinCALIS windows. Consider this example:

Presentation (Text) Window:

```
<TAB:1>one<TAB:3>six
<TAB:1>two<TAB:3>seven
<TAB:1>three<TAB:3>eight
<TAB:1>four<TAB:3>nine
<TAB:1>five<TAB:3>ten
```

Both the columns will be neatly aligned to the left, because of the <TAB:n> commands used.

Remember that basic Windows editing functions (Cut, Copy, and Paste) can make the insertion of commands like TAB quite simple.

TABbing

WinCALIS also supports tabbing with the Tab key in the ordinary word-processing sense. Tabs are inserted by pressing the Tab key in Author. Tabs are set at every 40 pixels. If desired, an author could also insert the equivalent text command <TAB:-1> in a script to tab over to the next tab stop. Note: this is not a reverse tab as supported by some word-processing applications. On a 640x480 VGA display, the right margin is equivalent to a little less than <TAB: 16> (allowing for window borders). <TAB:8> would center the cursor in the screen.

Note: Tab stops for the <TAB:n> command (put cursor at nth tab stop) in WinCALIS 1.x were set at 64 pixels, but are now set at 40 pixels, for WYSIWYG compatibility with Author. There is thus a 5:8 ratio between the new and old tab position. Old tab values in WinCALIS 1.x can be converted by multiplying them by 1.6 (8/5). E.g., the command <TAB:5> in an older script should be replaced with <TAB:8> for equivalent appearance. You can use the Find and Replace function in Author to convert old <TAB:n> commands, or if you used the <TAB:n> command frequently, you can define a keyboard macro (using Windows Recorder) to automate this conversion.

See Also: Placing Tabs

DELAyIng Execution

When elaborate portions of information are displayed in a window, it may be easier for a user to see that the information is built up gradually. This is possible when the author enters the <DELAY:n> command.

For example:

Presentation (Text) Window:

When reading a text for the first time, you should:<DELAY:1.75>

- focus on the general contents rather than details<DELAY:1>
- get a grasp of the structure by paying attention to the headings used <DELAY:1>
- paying attention to the title, which usually summarizes the contents of the text in a very brief way. <DELAY:1>

The first text line appears after which WinCALIS waits 1.75 seconds and then displays the second line. The third line will be displayed after one second. The other lines will follow, each after a one-second delay.

The <DELAY> command pauses WinCALIS at the point at which it is inserted. Time values down to 1/100th of a second can be used. If the <DELAY> command is used in a file which appears in a tutorial window (Alt+F[1-10]), it will only work the first time that the tutorial window is displayed. Afterwards, all text in the window will be displayed immediately. The <DELAY> command also can be inserted in a Feedback box in a Task menu Answers/Feedback dialog box. In that case, feedback will be delayed by the time specified by the author.

See Also: Time Delay

Sounding a BEEP

If an author wants to draw special attention to particular text, a short auditory signal may be included in a window or the Feedback box in a Task menu Answers/Feedback dialog box in WinCALIS Author. This is possible by means of the <BEEP:n> command. This command will make a beeping noise. The number of beeps can be set from 2 to 10. For example:

<BEEP:7>

will beep seven times.

Authors should be careful when using this command. If used frequently it has a great potential to annoy the user.

Sizing Windows

To suit your preference, you have the option of placing the Presentation (Text) Window either at the top part of the screen or at the bottom. One simple command at the beginning of the Presentation (Text) Window will tell WinCALIS where to place the window on the screen. The default setting is the Presentation (Text) Window at the top of the screen. To place it at the bottom of the screen, use <SIZE TEXT: LOW>. To place it again at the top of the screen, use <SIZE TEXT: HIGH>.

Syntax: <SIZE TEXT: [HIGH|LOW]>

Abbreviation: <T>

WinCALIS sets various defaults for the space a window takes up on your monitor screen. All windows are set to make use of the maximum horizontal (left to right) length of your screen. The height of your screen (top to bottom) is dependent on the type of window and the total number of character lines typed.

An author may wish to change the default window sizes set by WinCALIS. In order to do so, the <SIZE [W]:top, left, bottom, right> command must be used. The syntax for this command is:

<SIZE [W]:top, left, bottom, right>

In order to understand how to use the <SIZE [W]:top,left,bottom,right> command, you should be aware of the relation between the Size/Grid System and the <SIZE [W]:top,left,bottom,right> command and difference between the screen and a window. The screen is what you might call the total of the display capacity of your monitor. A window is a rectangular area on your screen in which information is displayed. The screen may be thought of as containing a grid of rows and columns from 0 to 100 for length and width. The grids are absolute to the screen itself and not to the particular window as in DOS-CALIS.

To size a window and locate it in the middle of the screen you enter the following parameter into the <SIZE> command:

<SIZE [W]:33,33,66,66>

The first parameter, “33” informs WinCALIS that the top of the window should be located 33% below the top of the screen. The second “33” causes the left border of the window to appear 33% away from the left-hand side of the screen. The first “66” parameter tells WinCALIS to locate the bottom border of the window 66% below the top of the screen. The last parameter causes the right-hand side of the window to appear 66% away from the left-hand side of the screen.

The [W] option allows you to define which window to apply the size to. If not specified, WinCALIS will apply the sizing to the window that you are currently editing in WinCALIS Author. For example, when you enter the <SIZE: top,left,bottom,right> in the Popup (Directions) Window, WinCALIS will apply the size to the Popup (Directions) Window.

Syntax: SIZE [W]:top,left,bottom,right>
No abbreviation

Window:	Syntax:	Abbreviation
Popup (Directions)	<DIR>	none
Interactive (Question)	<QUESTION>	<Q>
Presentation (Text)	<TEXT>	<T>
Tutorial	<F[1-10]>	none
Diagnostic Feedback	<D>	none

Note: The Lesson Description Window cannot be sized with the <SIZE> command. It is automatically sized to fit the text in it. The Diagnostic Feedback Window is also a special case. The D Window is created anew in every task and destroyed when the task is completed. So any command which references the D Window will have an effect only if the D Window currently exists and only for the current task. In order to size the D Window reliably, the <SIZE> command should appear in each D window for which the size is desired. One way of accomplishing this semi-automatically is to insert <SIZE> commands in the Script Options “Default correct feedback:” and “Default incorrect feedback:” boxes.

top	any number between 0 and 100
left	any number between 0 and 100
bottom	any number between 0 and 100
right	any number between 0 and 100

Using the SIZE Command in WinCALIS Lessons

Apart from the aesthetic value that sizing might bring, the <SIZE> command may have a functional use in some lessons.

Directions given at the beginning of the learning session may sometimes be easier to understand if the student can get a glimpse of the environment in which he or she will be working. Therefore an author can decide to reduce the size of the Popup (Directions) Window so that part of the Interactive (Question) Window can be seen by the student when reading the lesson directions.

The <SIZE> command could also be used to another purpose. Unless otherwise specified WinCALIS sets the window sizes to make use of the maximum horizontal (left to right) length of the screen. At times an author may want to use only the left-hand half of the screen to display text in the Presentation (Text) Window, while reserving the right-hand half of the screen for the Interactive (Question) Window. In that case the <SIZE T: 0,0,100,50> would have to be entered in the Presentation (Text) Window, and <SIZE Q 0,50,100,100> in the Interactive (Question) Window in order to get two column-shaped windows next to each other. (For a detailed discussion of Window Sizing in WinCALIS see the section CALIScript.)

A Note on Commands

There may be times when the commands in angled brackets appear literally in your script and do not function as they should. In order to remedy this problem, you must go back into the lesson via WinCALIS Author and insert a hard return (press the ENTER key) immediately before the first angled bracket that is appearing literally in the lesson. Also, there may be times when text does not fit entirely within the visible portion of a window. (This usually occurs when the FONTSIZE chosen is a large one, such as LARGE, XLARGE or HUGE.) As above, you should go back into the lesson and insert a return to break the text into visible chunks.

Note for advanced WinCALIS users: If a CALIS command in angle brackets gets spliced between two lines of script the part of the command which begins the new line is interpreted as text. This should not normally happen in WinCALIS Author. But if it does, you may use the Scratchpad Window and load the “bare” script by selecting the Document file: Open menu option from the File menu. After doing so you can select the file from the file list. WinCALIS Author will load the file in the Scratchpad Window. You may then edit the file and insert the necessary hard returns or do the necessary maneuvering to keep the command from being spliced. Refer to the section CALIScript for more information on editing 'bare' WinCALIS scripts.

Refining Feedback

In the section Question and Feedback Development you learn how to develop feedback in WinCALIS. As pointed out there, developing feedback information for the student is no more difficult than entering information in the appropriate feedback boxes. However, there may be times that you feel that the textual feedback in the Diagnostic Window, which usually appears at the bottom of the screen, is not sufficient, or not appropriate in all possible cases, etc. Therefore, WinCALIS has a whole range of extra options to make feedback to the student as specific and appropriate as needed.

Displaying the LAST Student Entry and the CORRECT Answer

At times it may be useful to repeat the last student entry in the feedback given to the student through the Diagnostic Window (or D-box). First consider this example:

Interactive (Question) Window:
Who is buried in Grant's tomb?

Correct/Continue Anticipated Response:
Grant

Feedback:
Very Good!

Anticipated Incorrect/Remain Here Answer:

{Lee|Lincoln}

Feedback:

<LAST> is in a tomb of his own!

If the student response to this question is “Lee” or “Lincoln,” it will be repeated to the student with the diagnostic message shown above.

As this example shows, the <LAST> command is handy for using the student's response within a diagnostic message that can be applied to more than one answer. It is used to print the student's previous answer to the screen. It is most commonly used in diagnostic messages for Incorrect/Remain Here student answers, but it may also be used in any other WinCALIS window. Each time the window is rewritten, the last student entry will be displayed.

When using this command, care must be taken to provide enough room for the longest anticipated student response. (No student response should exceed 99 characters and spaces.) Failure to observe this precaution can result in a diagnostic message disappearing into the right border of the screen if the student's last response and the message combined exceed 99 spaces. The student response can be limited using the <LENGTH> task option in order to avoid this problem. It will be discussed in WinCALIS Task Options in the section CALIScript.

Another way to solve the problem is to use the <LAST> command in a slightly modified way to limit the number of characters actually displayed to the student as feedback. You may use the command

<LAST: [length]>

in which [length] is the number of characters to be displayed to the student; it can be used to specify the number of characters of the student's response to display. For example:

<LAST: 4>

will display only the first four characters of the last student entry. If the student entry is longer than the defined number of characters to be displayed, WinCALIS will truncate the response; if the student entry is shorter than the defined number of characters to display, WinCALIS will add a sufficient number of blanks.

Combined with the <CORRECT> command, the <LAST> command can be particularly useful when you want a student to compare his own entry to the Correct/Continue answer. In such a case you would insert the following in the feedback box:

You answered: <LAST>

The correct answer is <CORRECT>

In this case the Diagnostic Feedback Window would show the above feedback message with the <LAST> command replaced by the last student entry, and the <CORRECT> command by the first Correct/Continue Anticipated Response for this task.

The <CORRECT> command can be used in other, more inventive ways. Consider the following Tasks:

TASK 1

Interactive (Question) Window:

How does the Gettysburg Address begin?

Correct/Continue Anticipated Response:

Fourscore and seven years ago

TASK 2

Presentation (Text) Window:

<CORRECT: #1> our fathers brought forth ...

Addressing Windows when Giving Feedback

In the example in the preceding section, Displaying the <LAST> Student Entry and the <CORRECT> Answer, the first correct answer to TASK 1 is used to begin the Presentation (Text) Window for TASK 2. The <CORRECT: #1> command is used to display a correct answer on the screen. If used by itself <CORRECT> displays the first correct answer from the task in which it appears.

<CORRECT n,#t>

Parameters:

n- the number of characters to be displayed from the first Correct/Continue answer
#t- the task label of the task of which the Correct/Continue answer should be displayed

It is possible to use the first Correct/Continue answer from another task by referring to it as

<CORRECT: #t>

The number of characters from the first Correct/Continue answer displayed by this command can be limited in this manner:

<CORRECT: n>

The two options for the <CORRECT> command can also be combined:

<CORRECT: n,#t>

For example: <CORRECT 4,#1> would print the first four characters from task 1. The # sign should be used in order to tell WinCALIS that what is following is actually a task label and not the number of the characters to display from the Correct/Continue answer. Be sure that Correct/Continue answers which will be displayed by the <CORRECT> command do not contain WinCALIS wild cards in order to avoid confusing the student.

Numeric values for <LAST> and <CORRECT> can also be set by the <SET> command. For example:

<SET: CORRECT 4>

will set the number of characters of the Correct/Continue answer to be displayed to (the first) four.

<SET: LAST 4>

will set the number of characters of the last student entry to be displayed to (the last) four. The syntax for these commands is:

<SET: LAST [length]>

<SET: CORRECT [length]>

For a full discussion of the <SET> command refer to the appropriate section in CALIScript.

The Window Commands

As stated previously, there may be times when specifying feedback in the Diagnostic Window is not sufficient. You might, for example, want to change the contents of the Presentation (Text) Window, highlight a portion of its text as a hint, add information to the Interactive (Question) Window, fill in a

blank when the students gives up or restore the window to its original state when the student decides to pass on a particular task.

All of this and more is possible through the Window Commands in WinCALIS. As with other commands they can be entered with your text in WinCALIS Author. This is a list of all the available Window Commands:

Window:	Syntax:	Abbreviation
Popup (Directions)	<DIR>	none
Interactive (Question)	<QUESTION>	<Q>
Presentation (Text)	<TEXT>	<T>
Tutorial	<F[1-10]>	none
Diagnostic Feedback	<D>	none

The [1-10] option in the <F[1-10]> command specifies which of the ten available Tutorial Windows is being addressed.

The Window Commands tell WinCALIS to activate the selected window, and insert text or perform designated commands until a Window Command activates another window. If you want to use one or more of the Window Commands following a student entry, you should insert them into the Feedback boxes of one of the WinCALIS Author Task lesson format options. You can insert them in any of the WinCALIS windows, but care should be taken to respect the flow of the script. For this reason we recommend that you refer to the section CALIScript, which discusses Window Commands extensively.

Suppose you would have anticipated the following Correct/Continue Answer in a particular task:

Correct/Continue Anticipated Response:

Grant

Feedback:

<DIR> Now you will be asked an easy question.

In the above example, when a student enters the correct answer "Grant," WinCALIS adds a line ("Now you will be asked an easy question") to the Direction Window, instead of displaying the message in a Diagnostic Window (or <D> box). In the above case the Popup (Directions) Window is shown only when the student presses F9 or selects the Directions option from the Options menu in WinCALIS.

Obviously, adding such a line would not be of much use to a student. You could think of other cases in which the Presentation (Text) Window Command would be very helpful in providing the student a contextual hint. For example in this task:

Presentation (Text) Window:

Upon his death, General <M: gr1>Grant
was buried in his tomb.

Interactive (Question) Window:

Who is buried in Grant's tomb?

Anticipated Incorrect/Remain Here Answer (Short Answer tasks):

Jones

Feedback:

<TEXT><CURSOR:gr1>Grant

<D>Incorrect! Maybe the underlined word in the Presentation (Text) Window might be of some help to you.

<TEXT><CURSOR:gr1>Grant

In this example the Presentation (Text) Window is addressed, as well as two other commands which tell WinCALIS exactly where the underlined text portion (Grant) should be positioned in the Presentation (Text) Window. The <MARK:x> command 'marks' a particular position in a text, so that

the <CURSOR:x> command can be used afterward to position the cursor correctly. (For a more elaborate discussion of the <MARK:x> and <CURSOR:x> commands, refer to the topics Positioning the Cursor and Text in Windows and Positioning the Cursor at Specific Marks in your Text or Question in this section.)

So, in the above example, the Incorrect/Remain Here answer “Jones” causes WinCALIS to go to Mark “gr1” in the Presentation (Text) Window (<TEXT>) and (over)write the word “Grant” in bold text. Next, the Diagnostic Window (<D>) is activated ('popped up') and the message “Incorrect! Maybe the underlined word in the Presentation (Text) Window might be of any help to you” is displayed. As with all D boxes (or Diagnostic Windows), WinCALIS will pause and the user will have to click the box or press the space bar in order to continue. As soon as the user has done so, the Presentation (Text) Window (<TEXT>) will be activated again, the <CURSOR:gr1> command will position the cursor at the exact position and the word “Grant” will be (over)written again, but this time in a common font appearance, thus restoring the Presentation (Text) Window to its initial state.

The same process defined “manually” in the above example can be carried out automatically using WinCALIS Author to generate cloze lessons. You can then choose Clue as an option from the Cloze Exercise dialog box to include contextual clues for a student. WinCALIS will generate a similar kind of underlying script, about which you will learn all in the section CALIScript.

Notice that in the above example, the <TEXT> command has been useful for rewording parts of the Presentation (Text) Window and saving other parts. In this way, information in the Presentation (Text) Window can be shared by several tasks, and modified according to need. The <T> or any other Window command can be inserted in the feedback boxes, and may be used in any other window. Each time it would complete the same function by telling WinCALIS to switch to the defined window and perform the designated commands. Finally, it should be noted that the <D> command, which activates the Diagnostic Window (or D-box), should not be used when you have not previously included any other Window Command in your feedback. Any information entered in one of the Task feedback boxes will be considered user feedback (and therefore be displayed in the Diagnostic Window or D box). Only when you first activate another window (for example, the Presentation (Text) Window, by using the <TEXT> command) should you use the <D> command to tell WinCALIS you want to switch back to the Diagnostic Window (feedback box). To conclude this section on Window Commands, we will add the following example involving the <QUESTION> command. It shows how WinCALIS manages to position the correct answer in a gap in a Cloze exercise, after the correct answer has been entered.

Interactive (Question) Window

<MARK:q1>_____ is buried in Grant's tomb.

Correct/Continue Anticipated Response:

Grant

Feedback:

Very Good!<QUESTION><CUR:q1>Grant

The correct answer “Grant” causes WinCALIS to replace the blank in the Interactive (Question) Window with the correct answer. It is possible to modify the Interactive (Question) window from another window by using the <QUESTION> command. Remember, as the student fills in a series of tasks the blanks in the text are replaced by the correct answers when they are entered. The Interactive (Question) Window remains otherwise unchanged.

MARK My Words: Positioning the Cursor at Specific MARKs in Your Text or Question

In the section Positioning the Cursor and Text in Windows you learned how to position the cursor in WinCALIS windows by using the <CURSOR:x> command. This command, which works with absolute (#r,#c) and relative (+#r,-#c) values, can be exploited fully and easily when it is used in conjunction with the <MARK:x> command.

The <MARK:x> command allows an author to direct a cursor to a point in a window without specifying the actual numerical values of the coordinates (the rows and columns). The author first “marks” a point in the window with the <MARK:x> command, giving the point a unique label. This point can be accessed with the <CURSOR:x> and <TOPLINE:x> commands. You will learn more about the <TOPLINE:x> command later in this section.

Syntax: <MARK:x>

Abbreviation: <M:x>

Parameter:

x- a unique label name

Changing the Look of Your Presentation (Text) Window

Presentation (Text) Window:

At the center of the computer system is the CPU which can be considered the 'brain' of the computer. Its main components are the microprocessor and the main memory. The speed and capacity of CPU's have been greatly improved with each new generation of computers. Each unit of the CPU is connected by <MARK:b1>a group of wires called a bus. There are data buses for carrying data from one unit to another, control buses for sending control signals to each unit and address buses for accessing each part of the main memory.

The main memory of a computer consists of RAM and ROM. RAM memory is volatile; when the computer is switched off the information contained in it will be lost.

Interactive (Question) Window:

What is the name of the group of wires that connect each unit of the CPU?

Correct/Continue Anticipated Response (Short Answer Task):

& bus

Feedback:

Correct, it is called a bus.

Anticipated Incorrect/Remain Here Answer (Short Answer Task):

&

Feedback:

<TEXT><CUR:b1><BLUE>a group of wires called a bus.<BLUE>

<D>Maybe the words highlighted in reverse blue could be of help.

<TEXT><CUR:b1>a group of wires called a bus.

In the above example the <MARK:x> command is used to provide a clue when the student enters an Incorrect/Remain Here answer. Instead of giving a plain feedback message in the Diagnostic Window (or D box), the attention of the student is directed towards the text. Technically spoken, it is done in this way: the first command <TEXT> found in the feedback box for the Anticipated Incorrect/Remain Here Answers activates the Presentation (Text) Window, the <CURSOR:b1> command directs the cursor towards the position in the text that was marked by the <MARK:b1> command. Next, the phrase “a group of wires called a bus” is written in underlined blue. The Diagnostic Window (D) is activated and will display the message “Maybe the words highlighted in underlined blue could be of help.” At this moment the execution of WinCALIS will pause until the user has clicked the message or pressed the Space Bar. After doing so, the user will notice that the Presentation (Text) Window is

restored to its original state; the Presentation (Text) Window (<TEXT>) is activated again, the cursor directed to the marked position (<CUR:bl>) and the phrase “a group of wires called a bus” overwrites the words in underlined blue.

More on MARKs

As you can notice, a mark is a point in a window created using the <MARK:x> command. The cursor can be moved to it using the <CURSOR:x> command as shown in the example above. The cursor is first directed to the Presentation (Text) Window using the <TEXT> command. It should be noted however that other windows such as the Interactive (Question) Window, Tutorial Windows, the Direction Windows can be accessed just as easily using the commands <QUESTION>, <F[1-10]> and <DIRECTIONS> respectively.

Marks are window-specific; a mark created in a window can only be accessed in that window. This also implies that different marks with the same name can exist in different windows. For instance, if a <CURSOR:x> or <TOPLINE:x> command is directed to a mark that doesn't (or does not yet) exist, the command is ignored.

Marks should not be given labels that conflict with the commands that access them. This means that the labels should not be the same as a window name, i.e. “Q” (Interactive (Question) Window), “DIR” (Popup (Directions) Window), “T” (Presentation (Text) Window) or “F” followed by a digit (Help Window), may not be used singly as mark names. Furthermore, marks should not start with a digit, a plus sign or a minus sign, and may not exceed eight characters. The <MARK:x> command is not seen by the student. In creating a script, however, it can take up a good deal of space if a number of <MARK:x> commands are used close to one another. Therefore, it is a good idea to first create the Directions, Text, Tutorial, and Interactive (Question) Windows which will appear on-screen to the student and then go back and place marks using the <MARK:x> command. This saves a lot of counting spaces or guessing when designing a script's on-screen appearance.

Changing the Look of Your Interactive (Question) Window

A mark can also be referred to when the cursor is in the same window in which the mark was created. To refer to a mark in a Interactive (Question) Window from the same Interactive (Question) Window is quite simple. The <CURSOR:x> command is used to move the cursor to the mark like this:

Interactive (Question) Window:

Here's a question on the text: <M:demo> Remember to read the above question carefully before answering!

<CURSOR:demo>What is the name of the central part of a computer?

The cursor is moved to the mark “demo” and the question “What is the name of the central part of a computer?” is inserted in the Interactive (Question) Window at that point. In order to refer to a mark created in one window from another window the cursor must first be directed to that window using a command such as <TEXT>, <QUESTION>, <F[1-10]>, <DIR>. To refer to the mark “demo” in the example shown above from a window other than the Interactive (Question) Window (where the mark was created) requires an additional command. The cursor must first be directed to the Interactive (Question) Window using the <QUESTION> command. Then the cursor can be directed to the mark. The text line below demonstrates this.

Presentation (Text) Window:

<QUESTION><CURSOR:demo>What is the name of the central part of a computer?

When using the <CURSOR:x> command in WinCALIS Author, you should be aware that WinCALIS Author itself will insert <CURSOR:x> commands in the actual script. For example, when generating a Short Answer task, WinCALIS Author will always take care of the final position of the cursor; it will

position the prompt to answer the question below the whole of the text in the Interactive (Question) Window. If you selected the CENTER Prompt Option from the Task Options dialog box, this prompt will be centered in the window.

MARKing for Hot Text

The <MARK:x> command is very useful for a variety of purposes. With it the author can set a point in a window and refer to it later in the script with the <CURSOR:x> command. However, it is also very useful in combination with the Special Response <MOUSE:x y> command. Suppose you want to include some help on difficult words in a text. In such a case the <MARK:x> command could be used in interaction with the <MOUSE:x y> command to make these words 'hot spots'. Hot spots are regions that can be clicked on with the mouse to execute a specified function. Consider this example:

Presentation (Text) Window:

At the center of the computer system is the <BLUE><M:c1>CPU<M:c2><BLUE> which is considered the 'brain' of the computer. Its main components are the <BLUE> <M:m1> microprocessor<M:m2><BLUE> and the main memory. The speed and capacity have been greatly improved with each new generation of computers. Each unit of the <BLUE><M:c3>CPU<M:c4><BLUE> is connected by a group of wires called a bus. There are data buses for carrying data from one unit to another, control buses for sending control signals to each unit and address buses for accessing each part of the main memory.

The main memory of a computer consists of <BLUE> <M:ra1>RAM <M:ra2> <BLUE>and<BLUE> <M:ro1>ROM<M:ro2> <BLUE>. RAM memory is volatile- when the computer is switched off the information contained in it will be lost.

Interactive (Question) Window:

<SET FC DKGREEN>Click the words in underlined blue to get a short definition.

<SET FC BLACK>What is the name of the group of wires that connect each unit of the CPU?

Correct/Continue Anticipated Response (Short Answer Task):

& bus &

Feedback:

Correct, it is called a bus.

Anticipated Neutral RE-EXECUTE Answers (Short Answer Task):

1. <MOUSE:c1 c2><OR><MOUSE c3 c4>

Feedback:

<BLUE>CPU<BLUE>: abbreviation for Central Processing Unit. It is the computational and control unit of the computer.

2. <MOUSE:m1 m2>

Feedback:

<BLUE>microprocessor<BLUE>: a central processing unit on a chip

3. <MOUSE:ra1 ra2>

Feedback:

<BLUE>RAM<BLUE>: acronym for Random Access Memory; semiconductor-based memory that can be read or written by the microprocessor or other hardware devices.

4. <MOUSE:ro1 ro2>

<BLUE>ROM<BLUE>: acronym for Read Only Memory; semiconductor-based memory that contains instructions or data that can be read but not modified.

Note: for adding Anticipated Neutral Responses to a Task Type other than Short Answer, see the section "Pass Through Window; A Tool for Advanced Authoring".

In the above example, clicking the words marked in underline and blue will cause WinCALIS to pop up a short explanation of the word. The words that may be clicked by the user are marked in blue for easy recognition. The explanation will be displayed in the Diagnostic Window, although this will not affect the score since the response type chosen in the Short Answer dialog box was Anticipated Neutral Re-execute Answer. It should be noticed that in this case the <MOUSE: x y> command requires two marks instead of one, the first mark being the upper left corner, the second, the bottom right corner of the hot spot. Remember, graphics can also serve as hot spots. You can create them using the same technique, although you may need to do some testing to find the bottom right position of the graphic since the graphic itself is not displayed in WinCALIS Author.

Using the menu option found under Format, Selection: Insert Marks, you can automatically place a marked pair around a selection of text. Author will place <M:x1> and <M:y1> as the first label pair of that authoring session, and increase sequentially for each subsequent pair.

Another example of marking for hot text:

In the Interactive (Question) Window:

You may click on any composer's name to hear an excerpt of a composition.

<M:x1>Bach<M:y1>
<M:x2>Beethoven<M:y2>
<M:x3>Chopin<M:y3>
<M:x4>Strauss<M:y4>
<M:x5>Rossini<M:y5>
<M:x6>DeBussy<M:y6>

Use Short Answer task type, Neutral (No Re-exec.) response type:
<MOUSE x1 y2>

Computer's Response/Your Feedback:
<AUDIO bach1> From Bach's c-minor fugue

Repeat this type of procedure for each marked pair anticipated response. You must remember that WinCALIS needs at least one Correct/Continue response before it will accept that as a complete task.

The Power of SPELLMARKing

Syntax: <SPELLMARK>

Synonym: <SPELLCHECK>

Abbreviation: <SPMK>, <SPCK>

WinCALIS has a unique way of checking spelling mistakes. Unlike a word processor, WinCALIS does not employ a dictionary-based approach to check the spelling of a particular student entry. Instead, it compares the student entry to the Anticipated Correct/Continue Response and reports in its own particularly efficient way.

From the Task Options and Script Options dialog boxes you can select the option Mark Spelling to include spell-marking when WinCALIS provides feedback an author didn't specify himself, but that is generated automatically by WinCALIS Author. However, you may find it very convenient to include the <SPELLMARK> command in the feedback you define yourself.

Interactive (Question) Window:

What do you call the thick-skinned nut that grows in exotic places and has a milk-like liquid in it?

Correct/Continue Anticipated Response (Short Answer format tasks):

coconut

Feedback:

Correct

Anticipated Incorrect/Remain Here Answer (Short Answer format tasks):

*c**

Feedback:

Check your spelling: <SPMK>

In the example shown above any answer beginning with the letter “c” will be compared by WinCALIS to the first Correct/Continue Anticipated Response and redisplay it with typographic markers to indicate particular spelling errors:

- Extraneous characters will be displayed as ~~strikeout~~ text.
- Missing characters will be displayed as an asterisk *.
- Wrong characters will be displayed with an underline.
- Transposed characters will be displayed in **reverse video** (foreground and background colors reversed).
- All four types of mistakes will be displayed in **bold** type.

The following student entries will be marked as:

Correct/Continue Response:	Student entry:	Marked as:
coconut	coconut	coconut
coconut	cocnut	coc*nut
coconut	cucunut	<u>cucunut</u>
coconut	cocount	coco unt

Furthermore, in those cases a student entry differs more than 50% from the Correct/Continue Anticipated Response, WinCALIS will display the first character of the Correct/Continue Anticipated Response, followed by a number of underscore characters () equal to the number of remaining characters in the word. For example:

Correct/Continue Response:	Student Entry:	Spell-check Feedback:
coconut	banana	c _ _ _ _ _

Note: WinCALIS takes capitalization into account when spell-marking unless you deselect the Check Case option from the Script or Task Options menu. In the latter case, WinCALIS will not mark the capitalization errors.

In cases where you defined an Correct/Continue Anticipated Response using wild cards, comparing the student entry against the Correct/Continue Anticipated Response, may make no sense. Neither does it make much sense if a student gives an answer which is completely different from the Correct/Continue Anticipated Response. In such cases WinCALIS will display the first letter of the Correct/Continue Anticipated Response.

SpellMark with Multiple Correct/Continue Anticipated Responses

For those tasks that have more than one Correct/Continue Anticipated Response, WinCALIS provides an automatic feature of Spellmark that automates feedback for “close-but-no-banana” student

responses. When a student inputs a response, WinCALIS compares it to all of the Correct/Continue anticipated responses (except those that have wild card characters embedded within), and gives a Spellmarked response based on the one that it is *closest* to. If it differs by more than 50% to all of them, then WinCALIS returns with the first character of each word in the *first* Correct/Continue anticipated response, followed by a number of underscores, representing the number of remaining characters in each word. For example:

Interactive (Question) Window:

Name ONE of the two tree types that were mentioned:

1. Correct/Continue Anticipated Response (Short Answer Task Type):

evergreen

2. Correct/Continue Anticipated Response

deciduous

This task will be handled as follows:

A response of “evergreen” or “deciduous” is considered satisfactory and will be returned with feedback. Any response that is *close* to either of the two target words will be marked accordingly.

Student entry:

evrgreen

dissiduous

oak

Marked as:

ev*rgreen

~~de~~ssiduous

e_____

Responding to Specific Input with the SpellMark Command

If you would like to include spell-mark information to the student that is appropriate to the answer he typed, you could do so in the following way by using a slightly modified version of the <SPELLMARK> command:

<SPELLMARK: Word>

Abbreviation: <SPMK: Word>

For example, suppose you would like WinCALIS to give specific spell-mark feedback in case the student types an answer such as “Sen Fransisco” to the question “What is the capital of the USA?” The common <SPELLMARK> command would, after comparing this entry to the Correct/Continue Anticipated Response (Washington), display “W_____” as spell-mark feedback. This doesn't tell the student much about the misspelling of the name “Sen Fransisco,” but gives rather a general hint about the correct solution. In such cases it would be wiser to include the following:

Interactive (Question) Window:

What city is the capital of the USA?

Correct/Continue Anticipated Response (Short Answer format task):

{Washington|Washington, {D.C.|DC}}

Feedback:

Correct, Washington is the capital of the USA.

Anticipated Incorrect/Remain Here Answers (Short Answer format task):

1. {{W|w}*n}

Feedback:

Washington is indeed the capital of the USA,
but check your spelling:<SPMK:Washington>

2. $\{\{W|w\}^*n\}, \{D.C.|d.c.|D.c.|d.C.\}$

Feedback:

Washington is indeed the capital of the USA,
but check your spelling:<SPMK:Washington, D.C.>

3. $\{\{W|w\}^*n\}, \{dc|Dc|Dc\}$

Feedback:

Washington is indeed the capital of the USA,
but check your spelling: <SPMK: Washington, DC>

4. *San Francisco*

Feedback:

Incorrect. San Francisco is a West Coast city,
famous for its Golden Gate Bridge, but it is
not the capital of the USA.

5. $\{\{S|s\}an \{F|f\}ranciscol\{S|s\}^*n \{F|f\}ra^*o\};$

Feedback:

Incorrect. San Francisco is a West Coast city,
famous for its Golden Gate Bridge, but it is
not the capital of the USA.

By the way you misspelled its name: <SPMK:San Francisco>

In this example, if a student misspells “Washington,” “Washington, D.C.” or “Washington, DC,” the mistake will be “caught” by one of the first three Anticipated Incorrect/Remain Here Answers. The feedback message will confirm that Washington is indeed the capital of the USA, but that the spelling is wrong. The spell-mark-marked entry will be redisplayed to the student.

In case the student types an answer such as “Sen Fransisco,” the feedback will contain remarks on both the spelling of the word, and on the fact that the answer given is not the correct answer to the original question.

It is obvious that the more anticipated Incorrect/Remain Here answers you specify, the more WinCALIS will be able to respond accurately and appropriately. Therefore we suggest that authors make an error analysis for the questions they wish to include in a WinCALIS script. This linguistic work will empower your scripts.

SPELLMARK with Multiple Correct/Continue Anticipated Responses

For those tasks that have more than one Correct/Continue Anticipated Response, WinCALIS provides an automatic feature of Spellmark that automates feedback for “close-but-no-banana” student responses. When a student inputs a response, WinCALIS compares it to all of the Correct/Continue anticipated responses (except those that have wild card characters embedded within), and gives a Spellmarked response based on the one that it is *closest* to. If it differs by more than 50% to all of them, then WinCALIS returns with the first character of each word in the *first* Correct/Continue anticipated response, followed by a number of underscores, representing the number of remaining characters in each word. For example:

Interactive (Question) Window:

Name ONE of the two tree types that were mentioned:

1. Correct/Continue Anticipated Response (Short Answer Task Type):

evergreen

2. Correct/Continue Anticipated Response
deciduous

This task will be handled as follows:

A response of “evergreen” or “deciduous” is considered satisfactory and will be returned with feedback. Any response that is *close* to either of the two target words will be marked accordingly.

Student entry:	Marked as:
evrgreen	ev*rgreen
dissiduous	de ssiduous
oak	e_____

FILLing Gaps (with Your Favorite Characters!)

Syntax: <FILL [length] [#task] [character]>
no abbreviation

Parameters:

length- any absolute number to indicate the number of spaces or characters you want to print

#task- the task label of the task of which the number of characters of the longest Correct/Continue Anticipated Response will be taken as the number of characters/spaces to print

Character- the character you want to fill the gap with

The <FILL> command is used to generate blanks. You will have noticed that in Cloze Exercises WinCALIS will automatically generate blanks for you. But there may be occasions in which you would like to insert blanks yourself in a text you created. The <FILL> command is a very powerful tool that you will certainly find easier than counting spaces or letters when you want to insert or change gaps in a text. For example, in its most commonly used form the command could look like this:

<FILL 10> is the capital of the USA.

In this case WinCALIS generates a blank 10 characters long. The character used will be the underscore (_).

If used by itself with no parameters <FILL> will create a blank as long as the longest anticipated correct response in the task in which it is used. Again, the blank is made up of underline characters (_).

As shown in the example above, <FILL> can be followed by a number to set the number of spaces in a blank to be generated, but the author can also refer to another task in the <FILL> command to generate a blank the length of the longest correct answer from that task. For example,

<FILL #3>

will generate a blank the length of the longest correct answer in task # 3.

The author can specify an alternate character to be used instead of the underline character as demonstrated below. For example,

<FILL 5 *>

will generate a blank made up of 5 asterisk signs (*****) instead of underscore characters.

You should keep a few rules in mind when using the FILL command.

- Digits (0-9) and upper ANSI characters may not be used with the <FILL> command.
- To use the characters: minus sign (-), comma (,), semicolon (;), colon (:), backslash (\), and space () as the fill character, precede the character with a backslash (\).
- The “greater than” character (>) cannot be used as a fill character. So, in order to create a blank of 5 absolute spaces, you would need to use the command <FILL 5 \ > while <FILL #3 \:> would generate a blank as long as the longest Correct/Continue Anticipated Response of Task 3, made up of colons (:).

The number of characters to be printed to the screen can be set to a constant length using the <SET> command. (You will learn more about the <SET> commands in another section of this section). <SET: FILL 10> will cause a blank of 10 characters to be generated when the <FILL> command is used. It can be changed with another <SET: FILL [length]> command. Here's an example how <FILL> can be used when developing an exercise:

TASK 1

Presentation (Text) Window:

```
<TAB:1><FONTSIZE FINE> <BO>WinCALIS <BO> is <TAB:1>a <M:x1><FILL 4 >1<FILL>
package that <TAB:1>was developed at Duke University. An authoring
<TAB:1>package is designed to create materials for <TAB:1>computer-aided instruction (CAI), which
<TAB:1>focuses on education, and for computer-<TAB:1>based training, which is <M:x2><FILL 5 >
<TAB:1>2<FILL 5 ><M:AP2>-specific and job-oriented <TAB:1>teaching. From the early stages,
<BO>WinCALIS<BO>
<TAB:1>was involved with foreign language teaching.
<TAB:1><BO>WinCALIS <BO> is a <M:x3><FILL 3 >
<TAB:1>3<FILL 3 > for Computer Assisted Language <TAB:1>Instruction System.
```

Interactive (Question) Window:

Choose the correct word for the first gap:

Correct/Continue Anticipated Response (Multiple Choice Question):

authoring

Feedback:

<T><CUR:x1>authoring

<D><CE>Correct

TASK 2

(Empty the Presentation (Text) Window by Select All and Cut from the Edit menu on the menu bar)

Interactive (Question) Window:

Choose the correct word for the second gap:

Correct/Continue Anticipated Responses (Multiple Choice Question format:)

application

Feedback:

<T><CUR:x2>application

<D><CE>Correct

TASK 3

Interactive (Question) Window:

Choose the correct word for the third gap:

Correct/Continue Anticipated Responses (Multiple Choice Question format:)

acronym

Feedback:

<T><CUR:x3>acronym

<D><CE>Correct

Only the Correct/Continue Anticipated Responses (Multiple Choice format) have been included in the above example for clarity's sake. The <FILL [length] [#task] [character]> command is used in the Presentation (Text) Window of the initial task. The blanks are preceded by a <MARK> command that can be referred to when defining feedback. The feedback consists of inserting the correct word and the message "Correct."

Choosing the TOPLINE of your Windows

Syntax: <TOPLINE [W] #r#m>

no abbreviation

Window:

Popup (Directions)

Interactive (Question)

Presentation (Text)

Tutorial

Diagnostic Feedback

Syntax:

<DIR>

<QUESTION>

<TEXT>

<F[1-10]>

<D>

Abbreviation

none

<Q>

<T>

none

none

Parameter:

#r- the absolute row number of a particular window

#m- a mark specified in a particular window

There may be times when your text will be longer than the actual window allows for. In such a case, WinCALIS will display a scroll bar at the right-hand side of the window and display the first line of a window as the top line of that window. If you refer to a portion of text that might not be on screen because the text is longer than what can be displayed at one time, you may find the <TOPLINE [W] #r> command helpful. Consider the following example:

Presentation (Text) Window:

<SIZE 0,0,30,100>At the center of the computer system is the CPU which is considered the 'brain' of the computer. Its main components are the microprocessor and the main memory. The speed and capacity have been greatly improved with each new generation of computers.

Each unit of the CPU is connected by <MARK:b1>a group of wires called a bus. There are data buses for carrying data from one unit to another, control buses for sending control signals to each unit and address buses for accessing each part of the main memory.

The main memory of a computer consists of RAM and ROM. RAM memory is volatile; when the computer is switched off the information contained in it will be lost.

TASK 1

Interactive (Question) Window:

Read this first paragraph and answer this question:

Which component is considered to be 'brain' of the computer?<TOPLINE T: 1>

...

TASK 2

Interactive (Question) Window:

Read this paragraph and answer this question: What is the group of wires called that connect each unit of the CPU?<TOPLINE T: 7>

...

In this example, the Presentation (Text) Window has been sized so that it can display on average one paragraph of the text at a time. For the first task, WinCALIS will always display the first line (<TOPLINE T:1>) of the text as the top line of the Presentation (Text) Window. For the second task, which holds a question about the second paragraph of the text, WinCALIS will display the sixth line (<TOPLINE T:6>) of the text in the Presentation (Text) Window.

From the example, you can see the <TOPLINE[W] #r> command is used to place a desired line within a window at the top of the window. Used by itself the <TOPLINE #r> command refers to the window in which it is used. The row must be specified.

<TOPLINE: 3>

displays the current window starting at line 3. As in the above example, the <TOPLINE> command can also be used from one window to refer to another window like this.

<TOPLINE T: 3>

displays the Presentation (Text) Window starting with line 3.

The cursor must always be available on-screen in a Interactive (Question) Window. Therefore a <TOPLINE [W] #r#m> command which directs WinCALIS to display a portion of the Interactive (Question) Window which does not include the current position of the cursor will be ignored.

The <TOPLINE [W] |#m> command can also be extremely useful when providing feedback to a student: For example:

Presentation (Text) Window:

<SIZE 0,0,30,100>At the center of the computer system is the CPU which is considered the 'brain' of the computer. Its main components are the microprocessor and the main memory. The speed and capacity have been greatly improved with each new generation of computers. Each unit of the CPU is connected by <MARK:b1>a group of wires called a bus. There are data buses for carrying data from one unit to another, control buses for sending control signals to each unit and address buses for accessing each part of the main memory.

Interactive (Question) Window:

What is the name of the group of wires that connect each unit of the CPU?

Correct/Continue Anticipated Response (Short Answer Task):

& bus

Feedback:

Correct, it is called a bus.

Anticipated Incorrect/Remain Here Answer (Short Answer Task):

&

Feedback:

<T><TOPLINE: 6><CUR:b1><BLUE>a group of wires called a bus. <BLUE> <D>Maybe the words highlighted in blue could be of help. <T><CUR:b1>a group of wires called a bus.

The <TEXT> command in the feedback box activates the Presentation (Text) Window and the <TOPLINE 6> command causes WinCALIS to display the sixth line of this window as the top line. In this way the second paragraph of the text will be displayed on top of the window, i.e. visible to the student. Next, the <CUR:b1> command, referring to a mark <MARK:b1> in the Presentation (Text) Window, positions the cursor at the right location to print the phrase "a group of wires called a bus" in blue. Finally, the message "Maybe the words highlighted in blue could be of help" is displayed, and

after pressing the space bar or clicking the message by the user, the original screen attributes are restored.

<TOPLINE T: mark>

will display the line in which the mark <M:mark1> occurs as the top line for the window. It is important to remember that, after using the <TOPLINE [W]: [#r|mark]> command for a particular task, you should use the command again if you want the first line of the following Task to be displayed as the top line. If not, WinCALIS will continue using the top line that was defined in the previous Task as the top line for a particular window in the following Task.

Note: The TOPLINE [W]: #r#m> command can also be used with a mark instead of a row number. In that case WinCALIS will display the line in which the mark occurs.

BLANKing and CLEARing Windows

In the section Question and Feedback Development you learned that you can blank windows by selecting the Blank option from the Task Options dialog box. WinCALIS will completely blank the window, and place the information for the next task in the window. This information can be the same as in the previous task if you decide not to change the contents of the Presentation (Text) Window from your initial task.

There may be occasions in which you would like to change a portion of the text in a window, without completely replacing the text. There also may be occasions when you want to replace a portion of the text in a window, depending upon the entry of the user. Such techniques are possible by means of the <CLEAR> command in WinCALIS. In WinCALIS terms blanking refers to erasing an entire window. You may insert the <BLANK> command at the top of any window, and the entire previous contents of a window will be erased. As stated before, you would normally use the Blank option from the Script, Script Options... menu or from Task Options in the Answers/Feedback Dialog Box and have WinCALIS do this for you automatically. If you would like to know more about the full scope of the <BLANK> command, please refer to the relevant part of the section CALIScript.

The <CLEAR> command erases parts of a window from the current cursor position on, to the right of the line or to the left of it. It will also erase the remainder of a line or the remainder of the text. There are three versions of the <CLEAR> command:

<CLEAR> clears text from the cursor position down to the bottom of the window

<CLEAR:LINE> clears text from the cursor position right to the end of the line

<CLEAR: LINE LEFT> clears text from the cursor position left to the end of the line

TASK 1

Presentation (Text) Window:

Mary had a little <M:aa>cow, its fleece was white as snow,

Interactive (Question) Window:

In the above line, there is one word that is wrong.

Type the correct word from the nursery rhyme.

Correct/Continue Anticipated Response (Short Answer task):

lamb

Feedback:

<T><CUR:aa><CLEARLINE><BLUE> <LAST><BLUE>;<D>Correct

Anticipated Incorrect/Remain Here Answer:

<T><CUR:aa><RED>cow<RED>

<D>The word in red is the wrong one, and should be replaced by another. <T><CUR:aa>cow

TASK 2

Presentation (Text) Window (no blanking):

<M:cc>But <M:bb> and everywhere that Mary went,
the lamb was sure to go,

Interactive (Question) Window:

In the above line there is one word too many. Type the word that should be left out.

Correct/Continue Anticipated Response (Short Answer task):

but

Feedback:

<T><CUR:bb><CLEAR LINE LEFT><CUR:,1> and everywhere that Mary went the lab was sure to
go <D>Correct

Anticipated Incorrect/Remain Here Answer (Short Answer task):

&

Feedback:

<T><CUR:cc><RED>but<RED>

<D>The word in red is not part of the nursery rhyme and should be left out. <T><CUR:cc>but

The above example shows how you can replace and leave out portions of text from a window following a student entry. The <CLEAR LINE> command in Task #1's Feedback box of the Correct/Continue Anticipated Response clears all text to the right of current cursor position on the line. The correct entry is inserted in the line by means of the <LAST> command. This is particularly useful in cases where there is more than one correct answer, or when the Correct/Continue Anticipated Response contains wild cards.

In the feedback for the Anticipated Incorrect/Remain Here Answer, the marks in the Presentation (Text) Window are used in another way they indicate the position where the cursor should be located in order to display the wrong word ("cow") in red, thus providing real contextual feedback to the student. In Task 2 the same principle has been applied. The difference, however, is that the line is not cleared to the right from the current cursor position on, but to the left, thus giving the opportunity to leave out the word "but," which was the superfluous word. The <CLEAR*> command may also be inserted directly into a window. In a case where an author includes two types of information in a Presentation (Text) Window (general information that is useful for all tasks; specific information that is useful for the current task) the specific information could be erased for each task in this way.

Presentation (Text) Window:

In the past many people who gained respect and celebrity during their lives chose to be buried in a tomb that would act as a monument in honor of them after their death. <MARK: aa> Christopher Wren, the famous British architect, preferred to be buried in one of his most famous buildings.

TASK 1

Interactive (Question) Window:

Where was Christopher Wren buried?

Correct/Continue Anticipated Response (Short Answer task):

St. Paul's Cathedral

Feedback:

Correct

<T><CUR: aa><CLEAR>

TASK 2

Presentation (Text) Window (no Blanking after Task 1):

<MARK:bb>Sometimes the place to be buried was not chosen by the deceased himself, but by the public. Napoleon Bonaparte was put to rest in a prestigious military building.

Interactive (Question) Window:

In what famous military building was Napoleon Bonaparte buried?

Correct/Continue Anticipated Response (Short Answer task):

Hotel des Invalides

Feedback:

Correct!

<TEXT><CUR:bb><CLEAR>

After the student gives the correct answer the cursor is moved to the mark “aa.” The <CLEAR> command erases the Presentation (Text) Window from that point onward. The procedure is repeated after each task. Notice how in the example above the <T> command is used to redirect the cursor to the Presentation (Text) Window where the <CLEAR> command is to be used.

Extra-CALIS Communication

The WinCALIS system is designed to allow for communication with files outside those normally used and created by its component parts, WinCALIS, WinCALIS Author, the Multimedia Editor, and the WinCALIS Printer Programs. For example, you may import text files to special tutorial windows in WinCALIS to provide a brief review right on the spot, as a student works through a lesson. Or, you might have a text passage in a computer file that you want to place in the Presentation (Text) Window without retyping it. WinCALIS has a simple facility for doing all of this. We'll start with the tutorials.

Creating Tutorials

In order to provide your students with on-line review and help for a particular lesson, WinCALIS Author is equipped with the ability to display up to ten text files of your devising as tutorials. For instance, if you are developing a lesson on past-tense verb conjugations, you might include a tutorial which will list various sample conjugations, and keep it available in case the student should need help. With the click of the mouse or push of a button, the student will be able to access your help. By selecting the Tutorials option in WinCALIS or pressing the Shift key along with the function key corresponding to the desired tutorial (Shift-F[1-10]), the student will open a Tutorial Window on screen that can be moved and resized so that it may be observed throughout the lesson.

Creating the tutorial files is quite simple. Using the Scratchpad Window in WinCALIS Author, just enter and arrange the tutorial information, and save it with an appropriate name in the same directory in which the lesson script was saved. By default, this will be \WINCALIS\SCRIPTS. Saving a tutorial file is a bit different from saving a WinCALIS script file. From the menu bar in WinCALIS Author, select File, then from the Document file section, select Save as.... The Export Active Window Contents dialog box will appear. Just enter the filename and select the appropriate directory if necessary. Check if the Save File as Type box displays the Unicode (*.*) selection. The file should be saved in Unicode format in order to be imported by WinCALIS.

In creating the tutorial file, you may incorporate many of the WinCALIS text formatting commands, such as <color> and the font sizing commands. Like always, enclose the commands in angled brackets.

Making the file available to the user is even easier than creating it. While in WinCALIS Author, simply click on the Task menu and select the Tutorial Files... menu item. The Tutorial Files Dialog Box will open, presenting a sequential layout of tutorial name boxes corresponding to the function keys

Shift + F1-F10. You should fill in the boxes with the names of the tutorial files you have created (or plan to create), being sure to include the extension (usually .TUT or .TXT), if there is one.

Note: Since WinCALIS converts all import files into binary distribution-version files with the extension .WCI, no matter what the original file extension, there may not be two tutorial files associated with a script with the same filename, but different extensions. E.g., there may not be two import files, one called "imp1.tut" and the other called "imp1.txt."

The full pathname is not necessary since the tutorial file should be in the \WINCALIS\SCRIPTS directory or the directory in which your script is stored. You may use as many of the ten tutorial boxes as you like. The tutorial files you defined for a particular task remain available to the student throughout the following tasks or until you decide to replace a particular tutorial in an F window with new contents. For each task you can have up to ten Tutorial Windows, therefore in a WinCALIS lesson script consisting of 5 tasks, you could have up to 50 different Tutorial Windows, with ten available for each particular task.

When you do replace one tutorial file by another in a following task, you should remember to place a <BLANK> command on top of your tutorial file. This command erases the contents of the previous tutorial file from the F[1-10] Window before the second tutorial file is imported. See the section CALIScript for more information about the <BLANK> command.

"Popping Up" Tutorials

Note: No window other than a Tutorial Window or the Popup (Directions) Window can be made to pop up in a lesson.

Syntax:

<POPUP [W]>

Abbreviation: <POP [W]>

[W] - Window Options:

F[1-10]- any Tutorial Window from F1 through F10

DIR- Popup (Directions) Window

The student may select a Tutorial at any time during the lesson, but there may be circumstances when you will want to display a Tutorial even when the student has not requested it. For example, if the student makes a particular error in responding to a question, then you might want to display a tutorial to clarify the point. In the feedback of a question involving possessive pronouns, you might call up a possessive pronoun tutorial as help. The command that you use is the <POPUP Fx> command. The x parameter refers to the tutorials assigned to function keys 1-10. You may use the <POPUP Fx> command from anywhere in the script, but it is most often used in negative feedback, i.e. after a student has answered incorrectly. Just enter the command in the feedback for a particular Incorrect/Remain Here answer. The Tutorial Window will then pop up whenever the student enters the Incorrect/Remain Here answer that was anticipated.

Instead of popping up a Tutorial Window (F[1-10]), you may wish to have the Popup (Directions) Window shown to the student again. This is possible by replacing the [Fx] parameter by DIR.

For example the command <POPUP F3> will display the third tutorial as if the student selected it with the mouse or keyed in Shift+F3. When a Tutorial Window is called for by a <POPUP F[1-10]> command, the student no longer has the option to move or resize the window. Closing the window must be done by pressing the spacebar or by clicking the mouse.

POPUP [W] NOLOCK

Syntax: <POPUP [W] NOLOCK >

Abbreviation: <POP [W] NOLOCK>

[W] - Window Options:
F[1-10]- any Tutorial Window from F1 through F10
DIR- Popup (Directions) Window

At times you may want to have a Tutorial Window pop up, but give the student the opportunity to move and resize the window freely and to decide to keep the window on screen while continuing the learning session or to close the Tutorial Window first before continuing.

This can be done by using a slightly modified pop-up command, <POPUP [W] NOLOCK >. This command will cause a window to pop up, and allow the window to be moved, resized, closed or kept on screen at any time. It works in much of the same way as if a student had selected the Tutorial Window from the Tutorials option or the Popup (Directions) Window from the Options option at the menu bar in WinCALIS.

IMPORTing Tutorial files

Syntax: <IMPORT [W] filename.ext>

Abbreviation: <IMP [W] filename.ext>

[W]- Window Options:

Window:	Syntax:	Abbreviation
Popup (Directions)	<DIR>	none
Interactive (Question)	<QUESTION>	<Q>
Presentation (Text)	<TEXT>	<T>
Tutorial	<F[1-10]>	none
Diagnostic Feedback	<D>	none

When a student opens a Tutorial Window in WinCALIS, he is actually taking text from a particular Unicode file (that you prepared using the Scratchpad in WinCALIS Author and saved as a Unicode file) and placing it in a window designated as Tutorial. The same can be done with any other window using the <IMPORT [W] filename.ext> command where XX.X is the DOS name of the imported file and [W] is the window to which the file is being imported. If no window is specified, the file is imported to the current window. [The filename may have any legal DOS name.]

If you have a separate Unicode file prepared with information for the Presentation (Text) Window, for example, you can save having to retype it by importing it into the Presentation (Text) Window. The same applies to the other windows as well. Here is an example of <IMPORT [W] filename.ext>:

Popup (Directions) Window:
Answer the questions by typing the full answer
At any time you may press
Shift+F1 to receive general help on WinCALIS
Shift+F2 to look at the WinCALIS keyboard
Shift+F3 to review the modal verbs
<F1>Here is some information on WinCALIS:
<IMPORT: help.bsc>
<IMPORT F2: keyboard>
<IMPORT F3: modals.gra>

This example sets up several tutorial windows for student reference; general help is available in the F1 window, a keyboard diagram is in the F2 window, and a chart of modal verbs is imported into the F3 window. Each of these windows may be relevant to many WinCALIS exercises. By placing each in a

separate file and importing them as needed, an author need not reproduce the same information in every script.

As you can see in the example, the <IMPORT> command imports the text files “help.bsc,” “keyboard” and “modals.gra” in appropriate Tutorial Windows. However, there is a slight difference between the way in which files are imported in the first Tutorial Window (F1) and the other Tutorial Windows (F2 and F3). In the first case, the Tutorial Window (F1) itself is activated first. The message, “Here is some information on WinCALIS” is first written to the F1 window. Next, the file help.bsc is imported in it. In the case of the other Tutorial Windows the command to import particular text files (“keyboard” and “modals.gra”) is given from the current active window (in this case the F1 window), since it was activated by the <F1> command. (see Window Activation Commands)

Files that are imported with the <IMPORT> command must be in Unicode text format and may include CALIS commands with the following exceptions and restrictions.

- <IMPORT> and <POPUP> cannot be included in imported files under any circumstances.
- Commands which redirect WinCALIS to a particular window including: <DIR>, <Q>, <T>, <F1> and <D> must be excluded from imported files.
- Commands which allow for the designation of another window including: <BLANK>, <CURSOR>, <CLEAR>, <MARK>, <SIZE>, and <TOPLINE> must not contain these designations in an imported file. The command must be directed to the current window (the window into which the file is being imported).

In general, include commands in an imported file that are designed to work only within the window to which the file is being imported.

Note: See CALIScript for these commands.

Like WinCALIS 2.0 scripts, files brought in with the <IMPORT> command must be in Unicode text format. Unlike CALIS scripts, they are not themselves executed by the WinCALIS program. They are brought just in for viewing by the student. Therefore, do not give your import files names with the extension “.cal” or they will displayed in your students' WinCALIS menu.

- No more than 50 instances of importing files with the <IMPORT tutfile> command can be processed by WinCALIS when it runs a *.CAL script file and converts it into a *.WCL file. This is due to the way in which WinCALIS checks time-date stamps to see whether it needs to convert new import files into *.WCI files during the script conversion process. There is no limit on the number of *.WCI import files which WinCALIS can handle when directly executing ordinary *.WCL script files.

Document File... Save and Save As...

As the inverse function of the import command, the Document File... Save As... option under WinCALIS Author's File menu allows you to create a Unicode file out of the information contained in the active window. It will be easy for you to use the Scratchpad Window, since its contents are not included in the *.WCA script database file when your work is saved with the Script.. Save option in the File menu.

Creating a separate information file (in Unicode format) is especially handy when you create a passage, such as a standard set of directions, that you will want to use again in later lessons. After you export the text in the window to a Unicode file, it is very easy to incorporate that text into another script using the <IMPORT> command.

For example, if you frequently use directions such as those below, you should save the file as Unicode while the Popup (Directions) Window is active and save it in your WinCALIS development subdirectory for easy importing.

Read the passage above and answer the following questions. You will have three tries to get the correct answer. You do have the option of passing through questions using the gray + and - keys, and you may give up after one attempt using the F6 key. Good luck!

If you save this passage as “c:\WinCALIS\scripts\direc” using the Document File.. Save As... option under the File menu, you may use the command <IMPORT: direc> in the Popup (Directions) Window of a new lesson to have the above directions appear when that lesson is run by WinCALIS.

See Also: The File Menu

Including Tutorial Information in Your Lesson Script

At times you may want to include tutorial information in your lesson script itself rather than using a separate file which is being read by WinCALIS. Including a tutorial in your lesson script can be done very easily in WinCALIS. Instead of using the Scratchpad Window in the WinCALIS Author program and saving the contents to a separate file in Unicode format, you just type the tutorial information in one of the main WinCALIS Windows. A window which is convenient for this purpose is the Popup (Directions) Window.

Popup (Directions) Window:

In this lesson you will be asked to answer the questions about the text. If you like, you can consult the glossary of difficult computer terms during the learning session by pressing Shift+F1.

<F1>Glossary of computer terms

RAM: acronym for 'random access memory'; semiconductor-based memory that can be read or written by the microprocessor or other hardware devices.

ROM: acronym for 'read only memory'; semiconductor-based memory that contains instructions or data that can be read but not modified.

microprocessor: a central processing unit (CPU) on a chip

CPU: abbreviation for 'central processing unit'. It is the computational and control unit of a computer

...

In the above example the top lines of the Direction Window contain information for the student on how to proceed through the lesson. It is also announced that a Tutorial Window containing a glossary of some computer terms is available by pressing Shift+F1. The actual information that should be included in the F1 Tutorial Window is preceded by the <F1> command. This command activates the F1 Tutorial Window and writes the information that is following in it. The Popup (Directions) Window is popped up at the beginning of the session, while the Tutorial Windows will be kept available for the student to consult them.

Naming Tutorials

Syntax: <TUTNAME F[1-10] description>

Tutorial Window Options:

F[1-10] - any Tutorial Window from F1 through F10

Text description- your description for the Tutorial Window, not to exceed 50 characters

In using Tutorial Windows, you will soon notice that often a descriptive name for Tutorial Windows such as “Irregular Verbs” or “Glossary of Computer Terms” is much more clear than “Tutorial 1” or “Tutorial 2,” which is displayed as a default by WinCALIS. Therefore you will find it convenient to name your Tutorial Windows by using the <TUTNAME> command.

Note that if you specify no Tutorial Window (F[1-10]), WinCALIS will assume that the name that you are defining is meant for the currently activated window. This implies that the <TUTNAME description> in this case must be preceded by a command that activates the Tutorial Window in question, i.e. <F[1-10]>. Consider the following example:

Popup (Directions) Window:

In this lesson you will be asked to answer the questions about the text. If you like, you can consult the glossary of difficult computer terms during the learning session by pressing Shift+F1. <F1>
<TUTNAME Glossary of Computer Terms> Glossary of computer terms

RAM: acronym for 'random access memory'; semiconductor-based memory that can be read or written by the microprocessor or other hardware devices.

ROM: acronym for 'read only memory'; semiconductor-based memory that can contains instructions or data that can be read but not modified.

microprocessor: a central processing unit (CPU) on a chip

CPU: abbreviation for 'central processing unit'. It is the computational and control unit of a computer

...
<TUTNAME F2 General Information><IMPORT F2: help>

Tutorial Windows F1 and F2 will display with the labels “Glossary of Computer Terms” and “General information” respectively. When the student pulls down the “Tutorials” menu, this text will also appear next to F1 and F2.

If the <TUTNAME> command is used within the text of an activated Tutorial Window, as is the case for the F1 Tutorial Window in the above example, the author does not have to specify which window is being named; the current one is assumed. Otherwise, if the <TUTNAME> command is used outside of the Tutorial Window, as shown in the case of the F2 Tutorial Window in the example above, the author must specify which Tutorial Window is being named or the command will be ignored by WinCALIS.

Note: In the case of the F2 Tutorial Window, the <TUTNAME> command must precede the <IMPORT> command. Any text or command after the <IMPORT> command will be ignored completely by WinCALIS.

...and More.

In this section you have learned how you can use the WinCALIS Author menu options and commands to customize WinCALIS lessons and how to have WinCALIS communicate with the user the way you want to. However, we have sufficed to give you an overview of the commands and options that you can use from within WinCALIS Author. As you become even more proficient, you will be able to manipulate the “task list.” This is a concept that remains completely invisible within WinCALIS Author. By using the <GLUE>, <ADD>, <DELETE>, and <INSERT> commands you will be able to add, delete, or insert tasks from the same or another WinCALIS Author-generated script. In order to do so, you should understand the sequential flow of a WinCALIS script. This is explained extensively in the section CALIScript. You can also learn more there about the WinCALIS Commands and Default Script and Task Options which the WinCALIS Author Program automatically generates for you.

Chapter 5:

The CALIScript Language

Thus far in your CALIS career, you have taken advantage of WinCALIS Author's easy-to-use format to develop pedagogically sound and exciting lessons. However, there is another level of lesson development, positioned behind the scenes in WinCALIS Author, which is the heart of CALIS instruction. This level is CALIScript, the CALIS authoring language, and it is ultimately responsible for all the communication between you and the student. For this reason, you should become at least somewhat familiar with CALIScript. With a fundamental grasp of the language, you will be able to enhance your lessons and explore some options not available through WinCALIS Author alone.

Note: These sections serve as an overview of CALIScript. They should help you understand how your WinCALIS lessons are implemented and allow you to make some improvements to your WinCALIS Author-developed scripts. They are meant to supplement the information presented about WinCALIS Author. By no means is it necessary to develop your lessons using CALIScript by itself. In general, you should use WinCALIS Author as much as you can for your purposes, then modify your lesson, if needed, using the versatile CALIScript. If you want further information on CALIScript, contact the Humanities Computing Center or its agents, as listed in the Introduction.

Role of CALIScript

As its name would indicate, CALIScript operates much like a simple computer programming language. It gives the computer instructions for displaying information, accepting input, accessing the mouse, reading and writing to disk drives, and a myriad of other functions. Fortunately, for the non-computer programmers of the world, CALIScript acts like an elementary “high-level” computer language, which means that it employs normal human language in a logical, easy-to-understand manner. Just like a very simple spoken language, anyone can learn to read and write in CALIScript, you just need to learn the words and the syntax. This chapter gives a rudimentary understanding of the language and some of its features which can enhance the lessons you create in WinCALIS Author.

Relationship Between WinCALIS and CALIScript

Before the development of WinCALIS and WinCALIS Author, authors created lessons in text-based CALIS using CALIScript exclusively. Since then, WinCALIS Author has *greatly* facilitated the production of lessons by actually writing the lessons in CALIScript for you. It takes the lesson you make in WinCALIS Author and translates it into CALIScript. This translation comes out in Unicode format as a *.CAL file and from there is read and executed by the WinCALIS program for the student. This type of file is known as a script because it lists the instructions, line by line, for interaction with the audience, i.e. your students. These instructions are played out as a kind of educational performance when the lesson is run.

In any computer process, elements of translation occur, taking the information units of the CPU, 1's and 0's, and turning them into understandable human language. Similarly, it is best to understand WinCALIS Author as a translation tool, bringing you from the highest, simplest windowed level to that of CALIScript written in the *.CAL file. From there, the information is further broken down to 1's and 0's for the computer, then rebuilt and enacted in human terms for the student. The bottom line for language education is that computer terms are universal and therefore provide a medium for effective communication and translation between our complex and diverse human languages.

Keep in mind that WinCALIS Author generates two new files when you use the Script *.WCA Save or Save as.. option from the File menu in WinCALIS Author. The first file with the extension .WCA is the file used by WinCALIS Author. It is the file you would open when you want to use WinCALIS Author to add, change or delete parts of a lesson. The second file that is generated with the extension .CAL, is the 'lesson' or 'text' file that will be used by WinCALIS. The lessons with the *.CAL extension can be selected from the Lesson Description Window in WinCALIS (when the option Text Conversion *.CAL from the Options menu on the opening screen is selected). When such a lesson is run for the first time, i.e. when a file with the extension .CAL is selected for the first time, WinCALIS will automatically generate a binary lesson file with the extension .WCL which will appear among the other *.WCL files in the Lesson Description Window (when the option No Conversion *.WCL from the Options menu on the opening screen from WinCALIS is selected). The advantage of a binary file format lesson *.WCL file is that WinCALIS can immediately execute or 'run' it, without having to perform any conversion. The *.WCL files cannot be edited. If you want to edit lessons, you can open the *.WCA files in WinCALIS Author. At a deeper level, you can also directly edit *.CAL files, by importing them as document file in the Scratchpad in WinCALIS Author.

.CAL File Structure

The overall structure of a *.CAL file is relatively simple. It consists of a list of tasks which can present information to the student, ask questions, call for student response and provide feedback. The tasks are arranged and assembled in a coherent order dictating the flow of the lesson. The tasks are further subdivided into statements, which determine the nature and format of the input or output for each segment of the task. When assembled, a script looks like a line by line dissection of the text of your lesson, peppered with special CALIS commands and formatted into statements.

Examine the sample script below, which shows an instance of the main components of a script. Every line of the script is actually a statement, so it is best to think of these hierarchic elements in terms of their relation to statements. The script itself and the individual tasks are just collections of statements, while the text and commands are just components of the statements.

You should note that each statement or line in a script is demarcated by a statement marker (explained below) indicating its type. Following the statement marker are the text of the statement, including wild cards, as discussed in Customizing and Communicating and in this section, and any commands. It is also important to note that commands are set apart from text by enclosing them in angle brackets. For example, the command <CENTER> tells WinCALIS to center a line of text, just as you learned in the section Customizing and Communicating that <RED> tells WinCALIS to turn the text red.

SAMPLE *.CAL SCRIPT

```
@<MAXTRIES:99><GIVE:1>
#0<NOSC><NOPR>
!Read the selection and answer the following questions.
?<ADD: 1--2>
#end

#1
/<BLA>On April 14, 1992, the American band
/called the Beastie Boys released their
/third record album to much critical
/acclaim. Within four hours, excited fans
/had purchased every available copy of
/the release in New York City.
?<BLA><CUR:1><CE>How many albums had
?the band released before April 14, 1992?
?<CE> <M:prompt><CUR:prompt>
```

```
?<TAB:3><M:a1><UL>a.<UL> none<M:a2>
-<MOUSE:a1 a2><OR><KEY: 'a'>;That is not quite right
?<TAB:3><M:b1><UL>b.<UL> 3<M:b2>
-<MOUSE:b1 b2><OR><KEY: 'b'>;That is not quite right.
?<TAB:3><M:c1><UL>c.<UL> 2<M:c2>
+<MOUSE:c1 c2><OR><KEY: 'c'>;
?<TAB:3><M:d1><UL>d.<UL> 1<M:d2>
-<MOUSE:d1 d2><OR><KEY: 'd'>;That is not quite right.
?<TAB:3><M:e1><UL>e.<UL> 4<M:e2>
-<MOUSE:e1 e2><OR><KEY: 'e'>;That is not quite right.
-&;Please type the letter of one of the choices listed
&above.
```

```
#2
?<BLA><CUR:2><CE>What is the name of the band?
?<CE> <M:prompt><CUR:prompt>
+Beastie Boys;
-Beastie Girls;
&Actually the members of the band are males.
-&;<CE><SPCK>
```

Reading and Writing *.CAL Files

As stated above, *.CAL files are Unicode files. Each character in them uses sixteen bits instead of the eight bits that are used by ASCII characters. The advantage of this is that WinCALIS can handle virtually every character of every language in the world. If you want to edit the lesson scripts you have produced at the lesson script level, i.e. directly in the *.CAL file, you can do so by loading it in the WinCALIS Author Scratchpad.

To Load a *.CAL file into the Scratchpad:

- Double-click on the Scratchpad icon at the bottom of the screen in WinCALIS Author. This will open the Scratchpad window to its default size, taking up about two-thirds of your screen.
- You may maximize the window by clicking the Maximize button at the upper right-hand corner of the window. (If you are not familiar with the Minimize and Maximize buttons in Windows applications, refer first to your Microsoft Windows manual.)
- Select the Open option under the Document file heading from the File menu. The Import into Active Window dialog box will pop up.
- Select the *.CAL file you want to import in the window. If you like, you can edit the *.CAL file now. You have all of the tools from the Edit menu at your disposal.

Authors who do not wish to use the Scratchpad window, but their own familiar ASCII editor (such as the Notepad editor in Windows) or word processor to edit the file, can do so by first 'translating' the imported file to ANSI/ASCII format. However, you should be aware that if your scripts contain exotic characters that are not part of the ASCII character table, these characters can not be processed in your ASCII editor or word processor.

To Translate a *.CAL file:

- Select the Document file: Save as.. option from the File menu. The Export Active Window Contents dialog box will pop up.
- Select the option ANSI/WinCALIS 1.1 from the Save File As Type list box and click the OK button. Your file will now be saved in a format (ASCII/ANSI) that can be read by all ASCII editors, and most word processors.

Note: The first time you run your lesson in WinCALIS, you have to select it from the Lesson Description Box displaying all of the *.CAL files in a particular directory. The *.CAL file list can be displayed when selecting the Text Conversion *.CAL option from the Options menu in WinCALIS.

If you want to start writing a script with an editor other than WinCALIS, you should follow the same format for converting it, but in the opposite direction. Suppose you used an ASCII editor or a word processor to write a simple lesson script and saved it with the extension .CAL. In order to execute the file in WinCALIS you would need to convert it in Scratchpad.

To import into the Scratchpad:

- If you used a word processor, you should first check that you saved it in ASCII format.
- Select the Document file: Open option from the File menu in WinCALIS Author, choose the file from the file list, and click the OK button. If the file is a true ASCII/ANSI format file, WinCALIS will, prior to importing it in the Scratchpad, display a warning message saying "This is not a WinCALIS 2.0 Unicode file. Opening it as a WinCALIS 2.0 file may cause problems."
- Select the Open file as ASCII/ANSI text button in order to import the ASCII file into the Scratchpad.
- Preparing the (imported/edited) file in the Scratchpad for use by WinCALIS is easy. Select the Document file: Save as option from the File menu. Again, the Export Active Window Contents dialog box will pop up.
- Check if the Unicode WC 2.0(*.*) option is displayed in the Save File as Type box, and make sure that you have given your file a filename with the extension .CAL. Click the OK button. Your file has now been saved in Unicode format, and is ready to be run in WinCALIS.

CALIScript Syntax

All CALIS commands, options, and special responses are composed of one or more keywords in capital letters, surrounded by opening and closing angle brackets (< >), and separated by one or more delimiters.

The backslash character \ allows students to use one of the actual characters assigned to the wild cards (^, [], *, |, {}, &), the semicolon (;) or the backslash (\) itself in their responses.

WinCALIS Statements

Statements and Statement Markers
General Rules for Statement Markers
Statements in Detail

Statements and Statement Markers

Each line of a CALIS script must begin with a statement marker. As a set of eleven special characters, the statement markers indicate what type of line is to follow and how WinCALIS will process it. Any line that does not begin with a statement marker will be ignored by WinCALIS. Each of the fourteen statement markers has a basic and unique function. They are listed immediately below and will be described in further detail later in this section.

C	Comment appearing in Lesson Description Window
K	Sets student keyboard
L	Sets default font
%	Variable declaration and initialization
@	Default settings for options
#	Task label
!	Popup (Directions) displayed to student
/	Presentation (Text) displayed to student
?	Interactive (Question) section presented to student
+	Correct/Continue response(s) anticipated
-	Incorrect/Remain Here response(s) anticipated
&	Continuation or “spillover” line
=	Neutral Re-execute response
\$	Neutral No Re-execute response

These statement markers form the backbone of all lesson development in CALIScript. Combined with your text and the CALIS keywords capitalized and enclosed in angle brackets--commands, special responses, and options--these markers compose the syntactic superstructure of any script.

General Rules for Statement Markers

With regard to the general use of these statement markers, there are a few general rules which apply to every script:

- Each line that is to be read by WinCALIS must start with a statement marker.
- Every task must contain a task label, marked by the # sign.
- Every task that requires a student response must contain both a task label and a Correct/Continue answer (+) statement.
- With the exception of the default task statement (@), the order in which statements appear in a task should follow the order illustrated above (see “CAL File Structure”). The label must be first, directions and text (if they occur) should be next, then the question itself, and (in)correct answer(s) each accompanied by feedback (if used).

Note: The third rule above makes a distinction between tasks that require a student response and those that do not. The student doesn't necessarily have to answer a question to complete a task. There are some tasks that an author can design (such as ones that set up a screen or give directions) that the student doesn't reply to at all. However, any task that does require a response must contain a Correct/Continue answer (+) statement.

At various times we will refer to something known as the “task list.” A task list is a list of tasks the student must complete before the lesson is finished. Unless specifically directed otherwise, WinCALIS will present the tasks in a script in the order in which they are written. When WinCALIS first loads a script, it looks for two special tasks: the zero task (#0) and the end task (#end). The zero task is a task whose label statement is #0. WinCALIS always interprets this as the start of the lesson.

The end task marks the end of the initial primary task list. It is a task whose label statement is #end. The end task is automatically the very last task executed in a lesson. The end task is not, strictly speaking, a part of any task list. It will not be scored, and should not, generally, ask the student a question to be evaluated.

The basic task list consists of the tasks “0,” “end,” and all tasks in between them. This can be modified only with the commands <INSERT>, <DELETE>, and <ADD>, which will be described in the following sections (See “Task List Manipulation Commands”).

At any point in the lesson, WinCALIS works with two task lists: the primary and the secondary task list. The primary task list contains tasks which the student is currently working on. The secondary task list is created using the <ADD> command. For now, we will concern ourselves with the primary task list and the statement markers.

The short script below illustrates the four basic functions of a WinCALIS script. For example, when the script has been prepared in the WinCALIS Author Scratchpad window and has been saved under the name example.CAL, it may look like this:

```
#0
!Fill in the blanks with the past tense of the verb.
!
!Example:
!I ___ at the movie yesterday. (to be)
!
!Answer: was
#1
?Question 1
?Bob ___ to the store. (to go)
+went
#end
```

WinCALIS will display the above script in two parts. First the Popup (Directions) Window will be shown with the directions (marked with the (!) symbol in the script file). The student continues by clicking or hitting ENTER. The student may also click the on-screen space bar. After the student has done so, the question in the task labeled #1, which is marked with the ? symbol, is presented. The computer waits for the student to answer. The student can bring back the Directions at any time by pressing the F9 key or by selecting the Directions option from the Options menu. The student can type the answer to the question: *went*. WinCALIS responds with a default feedback message: Correct. Unless the script is modified by the author, the next question will appear immediately after the preceding one.

Statements in Detail

The statement markers are described in more detail below. Italicized portions of the syntax are variable, to be determined by you, while the non-italicized portions must be included verbatim.

C The Comment Statement

Usage: *Ccomment*

The “C” statement marker is used to describe the contents of the script in which it is used. The text which follows this statement appears in the WinCALIS opening screen in the Lesson Description Window (or *Comment* Window).

In the Author Workstation version of WinCALIS, the Lesson Description Window in the WinCALIS opening screen has two different modes, which are not present in the ordinary Student Workstation version of WinCALIS. These are selected from the opening screen Options menu. WinCALIS is

installed during Author Workstation Setup so that it starts automatically with the Lesson Description Window in “Text Conversion (*.CAL)” mode. This mode is the most convenient for actual lesson development, during which the *.CAL text form of the lesson must be repeatedly converted to the *.WCL binary, distribution version. However, in order to view what the student sees in the Lesson Description Window, the “No Conversion (*.WCL)” option should be selected from the Options menu.

Note: You may change the default setting so that WinCALIS starts up on your Author Workstation in “No Conversion (*.WCL)” mode. Use the Windows Notepad, the DOS Editor, or other plain-text editor to edit the “ListBoxDisplay=” line in the [WinCALIS] section of the WinCALIS initialization file WINCALIS.INI, in the main WinCALIS program directory (usually the parent directory of the default scripts directory). The values are as follows:

ListBoxDisplay=0 (sets No Conversion *.WCL mode) and

ListBoxDisplay=1 (sets Text Conversion *.CAL mode).

The use of multiple C statements allows the author to format the appearance of the text within the Lesson Description Box. For example a C followed by a new line, also beginning with a C would create a blank line within the lesson description. If the use of multiple C statements creates a text too large to be displayed, at once a scroll bar will be displayed in the Lesson Description Box, so that the student can scroll through the text. If only one C statement is used, it is essential that no hard returns appear in the text. WinCALIS will automatically format the text as a single paragraph.

WinCALIS text attribute commands such as <BOLD> and <color> may be used within a C statement, but you should be aware that, if not properly switched off, they will also affect the descriptions of other lessons, when browsing through the Lessons Box.

C<GRAPHIC:ACCSCLOG.BMP><TAB:2>Grammar Lesson 1

C

C<TAB:2>The Irregular Verbs

C<CE>level: Elementary - EFL

#0

#1

...

Note: you can also include a graphic in the Lesson Description Window by using the <GRAPHIC: filename.bmp> command.

The Student Keyboard Statement

Usage: *Kkeyboard*

The “K” statement is used to indicate the character set which is first available to a student when beginning a lesson. For example:

KArabic-Xerox

will cause keyboard input to produce Arabic characters. The Arabic characters will be assigned to the keys according to the layout of the Arabic keyboard for the Xerox Star workstation. If the student selects the Keyboard Help option from the Help menu, the Arabic characters will be shown in their proper locations on the virtual keyboard displayed on screen.

When running the lesson, WinCALIS will ask at the beginning of the lesson: “Display Arabic-Xerox Keyboard?” When the student clicks the OK button, the keyboard help template will be shown. With the K: statement in effect, students may still open the Language menu and select another keyboard and character set. If the K statement is not used, the default keyboard/character set is the standard ANSI “English/W. Europe” one, or another “top” menu language installed for that workstation.

WinCALIS Author inserts the “K” statement at the top of the script when generating the *.CAL file. If the author clicks on the Script menu, selects the menu item “Script Generation Options,” and checks the option “Use [currently selected language keyboard] Keyboard” in the Script Generation Options dialog box.

There may be only one “K” statement in a script and it may occur anywhere in the script, but it will always take effect at the start of the script.

L The Default Font Statement

The optional “L” Default Font statement sets the default font for use throughout a script. There may be only one “L” statement and it may occur anywhere in the script, but it will always take effect at the start of the script.

In WinCALIS Author, the “L” statement may be inserted in the *.WCA Author database and generated as part of the *.CAL script file by typing it in the Pass Through Window. For example:

```
LArial
```

will cause the entire lesson to use the Arial font.

% The Variable Declaration Statement

The % statement marker introduces a variable declaration statement. Variable declaration statements may appear anywhere in the script.

All WinCALIS variables must be declared in a “%” variable declaration statement before they are used in the script. A variable can be “initialized” with a value at the time of declaration.

Variables can be numeric variables, used as counters or flags, or string variables, containing strings of one or more letters.

Names of numeric variables consist of letters and numbers. Here is a typical numeric variable declaration:

```
%index1=0
```

Names of string variables consist of letters and numbers, followed by a \$ dollar sign. Here is a typical string variable declaration:

```
%president$="Clinton"
```

See Also: Variables
Pass Through Window

The Default Statement

Default statements affect the behavior of WinCALIS for all following statements in the script, or until the default is changed. In general, the WinCALIS Author program allows you to set these options with the click of a mouse. Defaults may be changed anywhere in a script, but are set most frequently at the beginning of the entire script. Defaults may be changed within a script as often as necessary. Following the @ symbol, only recognized task options, enclosed within angle brackets (< >) will be executed. Examples of defaults which can be set with these statements include whether or not to score the questions asked of students; whether to have the questions scroll down the screen as they are answered or to have the screen blank as new questions appear; or whether or not to check for such things as correct capitalization and punctuation. These and other options are discussed in full in the section Task Options and Defaults.

The Task Label Statement

Each task must begin with a task label, which is simply the statement marker # followed by the *Tasklabel*, the unique name referring to the task itself. Normally, when you work with WinCALIS Author-created scripts, you modify sections within tasks which have already been created and named. Unless you are going to create a whole new task from scratch, you will not need to use the task label statement.

Note: Each task label must be unique. Do not use the same task label for two tasks in the same script.

There are few rules for giving task labels. The task label may be any numerical or alphabetical combination with a few exceptions. WinCALIS commands, or spaces are not acceptable. Furthermore, be aware that the task labels “0” (the number) and “end” are reserved to mark the beginning and end of the lesson. Names such as, #verb1, #7 and #dothisnow are acceptable. WinCALIS Author normally creates tasks with simple number labels, like 1, 2, 3, 4, etc., or in the case of cloze tasks, 1.1, 1.2, 1.3, 1.4, etc.

In the following script, the lines between #1 and #2, i.e. one question and one Correct/Continue Anticipated Response, form one task.

```
#1
?Bob ____ to the store. (to go)
+went
#2
```

In the following example, the special label “0” (the number zero, not capital letter “O”) tells WinCALIS where to start. This task commonly contains the initial directions for completing an exercise. The special label “end” is used to signal the end of the entire exercise. When WinCALIS executes the task “end,” it stops, gives the student the score, and returns to the lesson menu. This does not mean that the task “end” must always be written at the very end of a WinCALIS script. (For more information see the section Sequential Flow of the Script through Task List Manipulation, in which the task list is discussed in more detail.)

```
#0
! Fill in the blanks with the past tense of the verb.
! Example: I ____ at the movie yesterday. (to be)
!
! Answer: was
#1
?Bob ____ to the store. (to go)
+went
#2
?I ____ him an hour ago. (to see)
+saw
#end
```

The Popup (Directions) Window Statement

Usage: !*directions*

Directions fall under two categories in WinCALIS: Script Start-Up Directions and directions that are grouped with one or more questions. Because creating directions in WinCALIS Author is so easy it is unlikely that you will need to write them directly into a *.CAL file. If you do use the directions statement marker (!), all text after the ! will appear in the Popup (Directions) Window. During a lesson,

a student can call up the Popup (Directions) Window by pressing F9 or by selecting the Directions option from the Options menu in WinCALIS.

A set of directions may be as long as necessary. If the directions are longer than can be displayed by the Popup (Directions) Window at one time, a scroll bar (the bar over the right border of the Popup (Directions) Window) will be displayed to the user, allowing the student to scroll through all of the directions. Script Start Up Directions are displayed once at the beginning of the lesson. All other directions are linked with one or more questions; they will be displayed if the student hits the F9, show directions, key at the beginning of all tasks they are grouped with. For example, if you write instructions into the Popup (Directions) Window for task #1, write no directions for task #2 and #3 and then write a new set of directions for task #4, the directions you wrote for task #1 will be displayed for tasks #1,#2 and #3. The directions for task #4 will appear in task #4 and any subsequent task until a new set of directions is encountered. In other words, Directions #1 was grouped with tasks #1, #2 and #3.

/ The Presentation (Text) Window Statement

Usage: */text*

Like the directions statement marker, the text statement marker / is used to display a block of text in the Presentation (Text) Window to the student. Each line which sends information to the Presentation (Text) Window begins with the / marker.

The Presentation (Text) Window, like the Popup (Directions) Window, contains information the student will want to refer to when working through a WinCALIS lesson. While the initial directions appear once and then remain hidden (until the student calls for it), the Presentation (Text) Window remains on the screen, always within view of the student.

A Presentation (Text) Window, containing more information that can be displayed at once will automatically attach a scroll bar at the right-hand side of the window. The most common use of the Presentation (Text) Window is to present written material which the student will use to answer the questions asked. Presentation (Text) Windows are also useful for the display of further instructions, examples, or to show the student's progress. Consider this example:

```
#1
/Hello, Mary! How are you today?
/I am fine, thank you.
? Mary-----fine today.
+feels; Correct " Mary feels fine today" is the correct answer.
```

The Presentation (Text) Window is discussed further in the section Window Activation Commands.

? The Interactive (Question) Window Statement

Usage: *?question*

The question statement marker functions exactly like the text and directions statement markers, displaying the text of the question in the Interactive (Question) Window. Each line of the question text block must begin with the ? marker. This is an example:

```
#2
? Question 2
?
?Who feels fine today?
?   a. Mrs. J.
?   b. Mr. Z
```

- ? c. No one
- ? d. Mary

The Interactive (Question) Window will be discussed further in the section "Window Activation Commands."

+ The Correct/Continue Answer Statement

Usage: Correct/Continue Anticipated Response; feedback & spillover
Syntax: +*Correct Answer; Feedback & Spillover*

The Correct/Continue answer statement marker functions unlike any of the other statement markers discussed thus far because its implementation depends on the input of the student. After the + statement marker, the Correct/Continue answer is listed. Following the correct answer, feedback may appear, which will be displayed if that correct answer is entered by the user. This optional feedback is separated from the correct answer with a semicolon separator. If the feedback goes over one line in length, then the *Spillover* statement, described below, must be used.

Also unlike the other statements seen so far, the + statement may appear in several blocks in a task, thus giving you the option of indicating several Correct/Continue answers. By repeating the sequence of statement marker (+), Correct/Continue answer, semicolon (;), feedback, and spillover statements (&), you can anticipate a wide array of Correct/Continue answers.

Note: Every task which the student has to complete must have at least one Correct/Continue answer. If the student eventually "gives up" on the question, WinCALIS will display useful feedback. Unless the special <GIVEUP> response is present (discussed in the section "Special Responses"), the answer on the first Correct/Continue statement will be shown to the student. By default, WinCALIS will move on to the next question when the student chooses any of the available Correct/Continue answers. It is possible to require students to answer all of the available Correct/Continue answers before moving on. This is discussed in the section "Task Options and Defaults."

Much of the modification of WinCALIS Author-created scripts will come in the feedback portion of the task. Multiple Correct/Continue answer anticipation along with clue highlighting in non-CLOZE exercises are two frequent areas for improvement. Consider this example in which two Correct/Continue answers have been anticipated:

#2
?I ____ him an hour ago. (to see)
+saw; Very good. A correct answer is "I saw him
& an hour ago. "I did see him an hour ago" is also
& grammatically correct.
+did see; Very good. "I did see him an hour ago"
& is a correct answer. "I saw him an hour ago" is
& also correct.

- The Incorrect/Remain Here Answer Statement

Usage: -anticipated Incorrect/Remain Here answer; feedback & spillover
Syntax: -*Anticipated Incorrect/Remain Here Answer; Feedback & Spillover*

Anticipated Incorrect/Remain Here answer statements allow the author to provide helpful clues, review or reminders to a student who is on the wrong track. In terms of syntax, the anticipated Incorrect/Remain Here response statement is identical to the Correct/Continue answer statement except for its marker. Feedback is separated from the anticipated Incorrect/Remain Here response by a semicolon (;) and the spillover statement (&) must be used if the feedback goes over one line.

As you have seen, it is possible to write a working WinCALIS exercise that gives the default response “Correct” or “Incorrect” to any student answer. However, the difference between a merely functional and an excellent WinCALIS script is largely a matter of appropriate feedback, the responses that the computer (in truth, the author) makes to correct and Incorrect/Remain Here student input. Here is an example:

```
#2
?I ____ him an hour ago. (to see)
+saw; Very good. "I saw him an hour ago" is
& correct.
- seed; Incorrect. The verb to see has an irregular
& conjugation.
-sawed; Incorrect. The past tense form does not end
& in -ed.
```

= The Neutral Re-execute Response Statement

Usage: =neutral response;feedback &spillover
Syntax: =*anticipated neutral response;feedback &spillover*

The neutral re-execute response statement marker = is used in a WinCALIS task for student responses which are considered to be neither correct nor incorrect. In some ways, a neutral response is similar to an Incorrect/Remain Here response. The task is not completed by a neutral response as it would be by a correct response. Instead the task is redisplayed to the student (the author can provide proper feedback if necessary).

Unlike an Incorrect/Remain Here response, however, a neutral response is not counted in a student's score; in other words, it is not counted as an attempt to answer the current question.

This makes a neutral response ideal for making options available to students such as getting help for the current question or making a selection from a menu. Such choices are neither correct nor incorrect and therefore do not receive automatic feedback. Here's an example:

```
#1
/ Mr. B is self-employed.
? Question 1
? Press '/' to get an extra clue.
?
? Where does Mr. B work?
?   a. Mr. B works with Mr. J.
?   b. Mr. B is a lawyer.
?   c. Mr. B is unemployed.
?   d. Mr. B has his own business
+c;Correct
=<KEY '/'>; Actually, Mr. B is his own boss.
```

In the above example, pressing the / key will activate the neutral response line and display the clue “Actually, Mr. B is his own boss.”

\$ The Neutral No Re-execute Response Statement

Usage: \$neutral no re-execute response;feedback &spillover
Syntax: \$*anticipated neutral no re-execute response;feedback &spillover*

The neutral no re-execute response statement marker (\$) is used in a WinCALIS task for student responses which are considered to be neither correct nor incorrect.

The difference with the neutral response statement, however, is that instead of redisplaying the task to the student, WinCALIS will not rewrite the information in its windows. (Feedback can still be given to the student.) It is also possible to add information to the windows by means of the Window Activation Commands, which you can learn more about in the following sections (See the section Window Activation Commands.)

& The Spillover Statement

Usage:

+anticipated correct response; feedback
& spillover OR
-anticipated incorrect answer; feedback
&spillover OR
=neutral re-execute response; feedback
&spillover OR
\$neutral no re-execute response; feedback
&spillover

The spillover (&) statement allows for multiple lines of feedback for anticipated response statements. Feedback is the part of the lesson where students probably learn the most, so provision for comprehensive feedback is very important. Several spillover lines may be associated with the feedback portion of the answer statement immediately preceding it.

In the following example both the correct answer and the anticipated error are divided by a semicolon into the student's answer and the WinCALIS feedback response. It is only the WinCALIS feedback response that may spill over onto multiple lines.

?Bob ___ to the store. (to go) [past tense]
+went;Bob went to the store.
-goes;This is the present tense. We are looking for a
&one-word past tense form here. Hint:
&The correct answer begins with "w."

The window where the response appears is the D window (D for diagnostic feedback). Unless directed to the contrary, WinCALIS always places feedback in the D Window.

Note: The spillover statement works only with the feedback portion of anticipated Correct/Continue, Incorrect/Remain Here, or neutral response statements, not with the student response portion. This must be contained entirely on one line and not be over 99 characters in length.

Wild Cards



Introduction to WinCALIS Wild Cards and Error Analysis

In many card games, a special card is set aside as “wild,” and can act as a substitute for other cards. In WinCALIS, this “wild card” concept has been expanded to allow the use of a series of special characters as instructions that match many potential answers. WinCALIS wild cards may occur within the student answer portion of anticipated Correct/Continue, Incorrect/Remain Here, and neutral answer statements (i.e., the portion which appears before the ';' character).

In the section Customizing and Communicating you learn how wild cards can be inserted in the Anticipated Response boxes of the various task type dialog boxes. For your convenience, and because wild cards are best explained by example, and best learned by use, the WinCALIS wild card characters are described again here in reference to the script in CALIScript itself.

One of WinCALIS's most powerful features is the way in which it allows the author to anticipate a variety of possible Correct/Continue, Incorrect/Remain Here, and neutral answers for a question. The use of wild cards greatly simplifies the process of writing WinCALIS anticipated Correct/Continue, Incorrect/Remain Here, and neutral responses. It also gives the author flexibility in accepting or rejecting a range of responses.

By using wild cards the author can look for key words or phrases within a student answer without specifying exactly what the answer should be. This can save a great deal of time and also allows for manageable lessons.

WinCALIS compares the student response to the anticipated Correct/Continue, Incorrect/Remain Here, and neutral responses in the order in which they are listed in the script. When using wild cards be sure to list the answers with the fewest wild cards (the most specific ones) before the answers with the most wild cards (the most general ones). Both Correct/Continue and Incorrect/Remain Here responses may contain multiple wild cards.

Wild Cards Limited to Single Characters

^ The Caret

The ^ (caret) wild card matches any single character. Thus “an^” will match “and,” “any,” and nonsensical words like “anb” or “anc.”

[] The Square Brackets

The square brackets match any one of the characters found within the brackets-. Thus “an[dt]” will only match “and” and “ant” (but not “any”).

[-] The Square Brackets with Hyphen

If it is necessary to match one of a long sequence of characters, the author may use the hyphen (-) between the beginning and end of the sequence as a shorthand.

For example, the wild card [1-9] matches 1, 2, 3, 4, 5, 6, 7, 8 or 9, and the wild card [a-z] matches any small letter that occurs in English. In essence, WinCALIS looks for any characters which have a value that lies between and includes the values of the characters separated by the hyphen. Examples of such consecutive characters include the numerals 0 through 9, the capital letters A through Z, and the small letters a through z. For other characters, select the “Unicode Input” option in the Edit menu and browse.

If you wish to use the [] wild cards to match the hyphen character itself, you must:

- make the hyphen (-) character the first or last character between the brackets OR

- use the backslash (\) character (discussed in the section “Literal Use of Wild Card Characters in Student Responses”) before the hyphen.

Wild Cards Limited to Single Words

* The Asterisk

The asterisk wild card character * matches any sequence of letters in a word. It can be used at the beginning of a word (e.g. “*es” will match “bridges,” “foxes” and “sees” among other things), in the middle of a word (e.g. “*t” will match “treat” and “taught”), or at the end of a word (e.g. “an*” will match “and,” “another,” and “antidisestablishmentarianism”). In essence, the * wild card will match any character except the space character (), i.e. the character used to separate words.

Wild Cards Not Limited to Single Characters or Words

& The Ampersand

The ampersand wild card character (&) matches any sequence of words. Thus “& moon” will match “the man in the moon” as well as “the cow jumped over the moon.” The & wild card will match any character, including the space character () used to separate words. (Compare with the asterisk wild card (*), above.) This wild card is especially useful as the last anticipated Incorrect/Remain Here answer line as a catch-all for any possible Incorrect/Remain Here answer: -&; (feedback) is an anticipated Incorrect/Remain Here response for any possible student response (including typos).

Note: An ampersand (&) in the first column of a statement does not function as a wild card, but rather it signals WinCALIS to begin a spillover statement. The ampersand (&) is a wild card only if it is not the first character in a statement.

| The Vertical Bar

The vertical bar wild card character | separates explicit alternatives, any of which will match. The bar functions as an “or” operator. Thus “never|hardly ever” will match “never” and will also match “hardly ever” (but not “never ever”).

{ | } The Braces (Curly Brackets) with Vertical Bar

In cases where the vertical bar used alone would be ambiguous, braces are used in conjunction with the vertical bar(s) | to show how a series of anticipated responses are grouped.

“Mary {had a little lamb|, Mary, quite contrary}” will match either “Mary had a little lamb” or “Mary, Mary, quite contrary.”

Braces can be nested. For example:

+ {I {am|was} | We {are|were} {all|an} individual {s}}; Correct

Note: The last pair of curly braces contains only one character. This means: the letter “s” or no character at all. You should be aware that in this example the student can also type nonsensical answers which will be considered correct such as: “I was all individuals.” Remember though that students cannot see the listing of potential answers and are not likely to try arcane answer combinations. If you want to restrict the accepted responses to answers that make sense, you should construct more specific anticipated responses, probably by using more literal entries separated by vertical bars and enclosed in braces.

You may use wild cards inside of the curly braces.

Literal Use of Wild Card Characters in Student Responses

\ The Backslash Character

The backslash character \ allows students to use one of the actual characters assigned to the wild cards (^, [, *, |, { }, &), the semicolon (;) or the backslash (\) itself in their responses. For example:

+Jack \& Jill;(feedback)

will reply “Correct” and give appropriate feedback if the student enters the answer “Jack & Jill” (and only “Jack & Jill”), exactly as typed.

Note: Wild card characters used literally in anticipated responses, preceded by a backslash (e.g., *) usually must be preceded by two backslashes in WinCALIS 2.x (e.g., *).

The backslash should also be used in other portions of the script whenever an author wishes to display a WinCALIS reserved character to the screen. For example:

/This is the backslash character: \\
 /This is the ampersand character: \&
 /This is the semicolon: \;

Note in comparison that:

+Jack & Jill;(feedback)

will record “Jack and Jill,” “Jack and his girlfriend Jill,” and any other analogous response as “Correct.” WinCALIS considers the “&” here as a wild card, as it is not preceded by a backslash (\).

If you wish the students to use the backslash character \ in their responses, the same system as above applies: type the \ following a \. For example, +black\\white; (feedback) will reply “Correct” to the student response “black\white.”

This feature was designed for limited use in special circumstances. It is unlikely that you will need a student to type ^, [, *, |, { }, &, ;, or \ in a response very frequently. You may also refer to the sections “Using WinCALIS Wild Cards in Anticipated Responses” and “Sequencing Answers and Wild Card Use” in “Customizing and Communicating.”

WinCALIS Commands

Numerous special commands are available in CALIScript. Insertion of many of these commands in the WinCALIS script has been automated in WinCALIS Author and commands can be selected conveniently from several menus. In order to give you a complete overview of CALIScript, all of the commands are discussed in these sections.

Note: All CALIS commands, options, and special responses are composed of one or more key words in capital letters, surrounded by opening and closing angle brackets (<>), and separated by one or more delimiters.

Up to this point, however, we have dealt with statements, not with commands. Statements must be prepared properly for WinCALIS to operate properly. On the other hand, it is possible to prepare WinCALIS scripts without directly using any commands. These scripts may be created solely through WinCALIS Author. CALIS authoring language commands function to provide increased flexibility. They allow the author to create and customize an almost infinite variety of lessons. Commands included in WinCALIS fall into different categories:

- Basic presentation commands, which include special screen attributes, color use, font sizing, and a time-delay feature
- Window activation commands, which control movement between WinCALIS windows
- Window manipulation commands, which control the appearance and function of WinCALIS windows
- Advanced screen design commands, which control how windows appear on the screen
- Task list manipulation commands, which allow an author to adjust and reorder the task list
- Media commands, which activate an audio, animation or video clip, or display an image

This format is similar to that of the Special Responses (see section “Special Responses”) and Task Options sections (see “Task Options and Defaults”). Although some of the commands act in similar ways to special responses and task options, it is important to distinguish them properly.

Commands may occur only in text being sent to some window. This includes the Popup (Directions) Window (!), the Presentation (Text) Window (/), the Interactive (Question) Window (?), the Lesson Description [or Comment] Window (C), the Tutorial Windows (F1-F10) and the feedback portion of anticipated Correct/Continue (+), Incorrect/Remain Here (-), and neutral (= or \$) response statements and spillover (&) lines, which is being sent to one of the WinCALIS windows as a response to some particular student entry.

- Commands should be excluded entirely from task label (#) and default (@) statements.
- Special Responses may occur only in Correct/Continue, Incorrect/Remain Here, and neutral response statements, and only in the student answer portion--viz. before the separator (;).
- Task Options may occur only in task label (#) and default (@) statements.

The task list manipulation commands <INSERT>, <ADD>, <DELETE>, <RANDOM>, <GLUE> and <UNGLUE> are covered separately in the sections Sequential Flow of the Script through Task List Manipulation and Implementing a WinCALIS Task Database.

Basic Presentation Commands

The Basic Presentation commands include commands for setting screen attributes (the color or appearance of text on the screen) and a time-delay function.

Text Attribute Commands

Text attributes in WinCALIS are specified with the following commands.

<BOLD>
<ITALIC>
<REVERSE>
<STRIKEOUT>
<UNDERLINE>

The above text attribute commands are used to switch the attribute on and off. Take, for instance:

?Who was the architect of the <BO>Eiffel tower<BO>
?in Paris?

The first command switches the bold typeface on, the second appearance of the command switches it off. The other commands function in the same way. If the author wishes to make an entire line appear in a particular attribute type, the command need appear only once at the beginning of the line.

The text attribute commands do not work for multiple lines. In the event that more lines should be displayed in a particular type, the <SET [W] ATTRIBUTE attribute> commands (see More New Features of CALIScript) may be used (except for reverse video); or the text attribute command can be inserted in all of these lines.

<BOLD>

Abbreviation: <BO>

#sample
?Who was the architect of the <BO>Eiffel tower<BO>?
+Eiffel; Correct!

The words “Eiffel tower” appear in **bold** type. The <BOLD> command is used to highlight text and give variety to the appearance of the screen.

<ITALIC>

Abbreviation: <IT>

#sample
?What city hosted the 1992 <IT>Olympic Games<IT>?
+Barcelona; Correct!

The words “Olympic Games” appear in *italic* type. The <ITALIC> command is used to highlight text and give variety to the appearance of the screen.

<REVERSE>

Abbreviation: <RV>

#sample
?What is the name of the capital of
?<RV>China<RV>?
+Beijing; Correct!

The <REVERSE> command is used to display characters in **reverse video** (foreground and background colors reversed), including blank spaces.

<STRIKEOUT>

Abbreviation: <SO>

#sample
?What country is <SO>Berlin <SO> the capital of?
+Germany; Very Good!

The <STRIKEOUT> command is used to display characters in ~~strikeout~~ typeface (including blank spaces).

<UNDERLINE>

Abbreviation:

```
#sample
?In what country is the <UL>Parthenon <UL> seen?
+Greece; Very Good!
```

The <UNDERLINE> command is used to underline characters (including blank spaces).

Text Color Commands

The following colors are available in WinCALIS through the use of color commands:

black, blue, dark blue, brown, cyan, dark cyan, dark gray, light gray, green, dark green, light green, olive, orange, pink, peach, purple, dark purple, red, dark red, white, yellow, and light yellow.

The color commands in WinCALIS are the following:

Command	Abbreviation	Color
<BLACK>	<BLK>	Black
<BLUE>		Blue
<BROWN>		Brown
<CYAN>		Cyan
<DKBLUE>		Dark Blue
<DKCYAN>		Dark Cyan
<DKGRAY>		Dark Gray
<DKGREEN>		Dark Green
<DKPURPLE>		Dark Purple
<DKRED>		Dark Red
<GREEN>	<G>	Green
<LTGRAY>		Light Gray
<LTGREEN>		Light Green
<LTYELLOW>		Light Yellow
<OLIVE>		Olive
<ORANGE>		Orange
<PEACH>		Peach
<PINK>		Pink
<PURPLE>	<PUR>	Purple
<RED>	<R>	Red
<WHITE>	<WHI>	White
<YELLOW>	<YEL>	Yellow

Note: DK is the abbreviation for DARK and LT stands for LIGHT.

```
#sample
?Who is buried in <BLUE>Grant's<BLUE> tomb?
+Grant; Very Good!
```

The word “Grant's” appears in blue type.

The color commands are used to produce colored text and give some variety to the appearance of the screen. Like the Text Attribute commands, they are used to switch color use on and off.

If an author wishes to make an entire line appear in red type, the <RED> command need only appear once, at the beginning of the line. If only one word is to appear in color within a line, then the color attribute must be switched on and off by a color command before and after the word.

Text attribute commands such as <BOLD> and <UNDERLINE> can be combined with <color> commands as shown below.

?<BO><RED>This will appear in bold red type!

See Also: Color Me CALIS

Setting Permanent Colors

If an author wishes all the text in a particular window to be of one color, he may use the <SET: [W] FCOLOR color> command. By using the <SET: [W] FCOLOR color> command, the window's total appearance is specified for as long as it is on the screen or until it is changed with another command. Refer to the section Advanced Screen Design Commands for a full discussion of the <SET> command.

Note: Window names may be abbreviated.

<SET: [W] FCOLOR color>

Abbreviation: <SET: [W] FC color>

Options: Window or [W]

Window:	Syntax:	Abbreviation
Popup (Directions)	<DIR>	none
Interactive (Question)	<QUESTION>	<Q>
Presentation (Text)	<TEXT>	<T>
Tutorial	<F[1-10]>	none
Diagnostic Feedback	<D>	none

Color Options: Color

WHITE	LTGRAY	DKGRAY
PURPLE	DKPURPLE	BLUE
DKBLUE	CYAN	DKCYAN
OLIVE	LTGREEN	GREEN
DKGREEN	LTYELLOW	YELLOW
RED	DKRED	PINK
PEACH	BROWN	BLACK

For acceptable color abbreviations, see Text Color Commands.

The use of the Window parameter ([W]) is optional. If not specified, WinCALIS will apply the text color to the current window.

```
#sample_1
/<SET FC GREEN>De gustibus et coloribus non
/est disputandum.
/(Des couleurs et des goûts on ne discute pas)
?<SET FC BLUE>What should not be argued
?about?
```


+colors and tastes; Correct

In the above example the entire text in the Presentation (Text) Window will be displayed in green, while the question in the Interactive (Question) Window will be displayed in blue. All new text added to or replacing the previous text in the Presentation (Text) Window will be displayed in green until the text color is changed by a new command. Since no window is specified, WinCALIS applies the <SET FC GREEN> and <SET FC BLUE> color commands to the windows in which they are inserted.

#sample-1

/<SET FC GREEN>De gustibus et coloribus non

/est disputandum.

/(Des couleurs et des goûts on ne discute pas)

#sample-2

?<SET FC BLUE>What shouldn't be argued about?

+colors and tastes; Correct

-collars and gusts; Incorrect <SET T FC RED>

Note: The Presentation (Text) Window needs to be 'isolated' (in a separate task #sample-1) from the task in which the question is displayed (#sample-2). If not, WinCALIS would return to the first line of the task and execute all of the statements following, including the <SET FC GREEN> in the Presentation (Text) Window, switching the text color back to green, because "collars and gusts" is defined as an Incorrect/Remain Here response (-).

In the previous examples the <SET T FC RED> will cause the text in the Presentation (Text) Window to be displayed in red, in case the student answers "collars and gusts." The text color in the Interactive (Question) Window remains unchanged since the <SET T FC RED> command, used only in the feedback portion, refers to the Presentation (Text) Window (T). Note again that the Window option may be abbreviated. For a full discussion of the Window Activation commands, please refer to the section Window Activation Commands.

An author wishing to set the background color to one particular color for the full window can do so by using the command: <SET: [W] BCOLOR color>. Again, by using this command, the window's total appearance is specified for as long as it is on the screen or until it is changed by another command.

<SET: [W] BCOLOR color>

Options: Window Options ([W]):

Window:	Syntax:	Abbreviation
Popup (Directions)	<DIR>	none
Interactive (Question)	<QUESTION>	<Q>
Presentation (Text)	<TEXT>	<T>
Tutorial	<F[1-10]>	none
Diagnostic Feedback	<D>	none

Color Options:

WHITE	LTGRAY	DKGRAY
PURPLE	DKPURPLE	BLUE
DKBLUE	CYAN	DKCYAN
OLIVE	LTGREEN	GREEN
DKGREEN	LTYELLOW	YELLOW
RED	DKRED	PINK
PEACH	BROWN	BLACK

For acceptable color abbreviations, see Text Color Commands.

#sample

```
/?<SET BC BLUE><SET FC WHITE>White  
/on blue...  
?<SET BC WHITE><SET FC BLUE>or blue  
?on white?  
?What do you prefer?  
+blue on white;<SET T BC WHITE><SET T FC BLUE>  
+white on blue;<SET Q BC BLUE><SET Q FC WHITE>
```

The Presentation (Text) Window is initially displayed in white characters on a blue background, while the Interactive (Question) Window is displayed in blue characters on a white background. A student entering “blue on white” as a response will cause WinCALIS to display the Interactive (Question) Window in blue characters on a white background. Answering “white on blue” will generate the opposite. Because both answers have been defined as “correct,” WinCALIS will execute the commands that were defined as feedback, and move on to the next task.

See Also: Setting Foreground and Background Colors

Setting Temporary Font Sizes

Another typographic attribute that an author may consider to be useful, is the ability to size (portions of the) text. As with the Text Color commands, an author has the option to make the settings temporary or permanent.

Changing the font size of a portion of text is done by using the <FONTSIZE size> command. The command provides seven different font sizes to choose from.

<FONTSIZE size>

Abbreviation: <FS size>

Size Options (size):

TINY
FINE
SMALL
MEDIUM
LARGE
XLARGE
HUGE

For example:

```
#sample  
?What is your font size:  
?<FS SMALL>small,<FS MEDIUM> medium  
?<FS LARGE>large, or <FS XLARGE>extra large?
```

In the above example, the words “small,” “medium,” “large” and “extra large” are all displayed in a font size that is matching to the words themselves. An author can use font sizes to focus on portions of the text, or as a layout tool to make lessons more attractive. The <FONTSIZE size> command will only work for the current line of text or until another <FONTSIZE size> command is encountered.

See Also: Sizing Up Your Text

Setting Permanent Font Sizes

Setting a font size permanently, i.e., for the text of a particular window, is possible by means of an alternative command:

<SET: [W] FONTSIZE size>

Abbreviation: <SET: [W] FS size>

Options: Window or [W]

Window:	Syntax:	Abbreviation
Popup (Directions)	<DIR>	none
Interactive (Question)	<QUESTION>	<Q>
Presentation (Text)	<TEXT>	<T>
Tutorial	<F[1-10]>	none
Diagnostic Feedback	<D>	none

Size Options (size):

TINY
FINE
SMALL
MEDIUM
LARGE
XLARGE
HUGE

```
#sample
?<SET FS SMALL>What is your fontsize?
?(1) small, <FS MEDIUM>(2) medium
?<FS LARGE>(3) large, or <FS XLARGE>(4) extra large?
?Type the number of the option you prefer.
+1;
+2;<SET Q FS MEDIUM>
+3;<SET Q FS LARGE>
+4;<SET Q FS XLARGE>
```

Note: In the Interactive (Question) Window the 'default' fontsize, which has been set by the initial <SET FS SMALL> command, is resumed after the designated text is temporarily displayed in font size "extra large," because the temporary command only works until the end of the line in which it is inserted.

In the example, pressing 1 will cause WinCALIS to move on to the next task, without making changes to the font size used in the Interactive (Question) Window. However, pressing 2 will set the font size for the entire Interactive (Question) Window to 'medium', except for portions of the text for which the font size has been changed by means of temporary <FONTSIZE size> commands.

See Also: Sizing Up Your Text

Centering Text On-Screen

WinCALIS allows you to center text on-screen automatically. The <CENTER> command works very much like the text attribute commands, except that it can not be switched on and off. All the text placed after the <CENTER> command in a particular line will be centered on the screen.

<CENTER>

Abbreviation: <CE>

The <CENTER> command must be used once for each line of text to be centered. It will work in all windows. For example:

```
#sample
!<CE>This line will appear centered on-screen.
```

The command is very useful for two reasons. One: moving text into the center of the screen generally enhances lesson appearance. And two: it is not necessary to increase the file size of scripts to format their on-screen appearance with space characters.

Placing Tabs

Another command an author will appreciate while designing a lesson screen is the <TAB: n> command. It allows you to place tabs, and, by doing so, to align text to the left.

<TAB: n>

no abbreviation

Parameter: n an absolute value; the highest value depends on the fontsize that is used.

For example:

```
#sample
!<TAB:1>Both of these lines will appear on-screen
!<TAB:1>starting at the tab 1.
```

Notice that the parameter used with the <TAB: n> command, is absolute. The <TAB:n> command does not work cumulatively. For example:

```
#sample
!<TAB:1><TAB:1><TAB:1>Directions
```

will position the word “Directions” at tab stop 1, although three <TAB: 1> commands were used. If you would like to position the word “Directions” at the tab stop 3, you should use this command instead:

```
#sample
!<TAB:3>Directions
```

Also note that the <TAB: n> command is a handy tool to create neat-looking columns in WinCALIS windows. Consider this example:

```
#sample

/<TAB:1>one <TAB:3>six
/<TAB:1>two <TAB:3>seven
/<TAB:1>three <TAB:3>eight
/<TAB:1>four <TAB:3>nine
/<TAB:1>five <TAB:3>ten
```

Both the columns will be neatly aligned to the left, because of the <TAB:n> commands used.

Time Delay

<DELAY: n>

no abbreviation

Parameter: n, time in seconds, using up to 2 decimal places

The <DELAY: n> command delays the operation of WinCALIS for n seconds before continuing. The command is used to pause WinCALIS at the point at which it is inserted. Delays can be set to within 1/100 of a second. Only the Anglo-American convention of using a period (.) to indicate a decimal place is supported by WinCALIS. Do not use a comma (,) For example:

```
#sample
?Who is buried in Grant's
?<DELAY:2.75>BACKYARD?
```

The first question line appears after which WinCALIS waits 2.75 seconds and displays the second line. The <DELAY: n> command can be used in the Directions, the Text, the Question, and the Diagnostic Feedback Window (D Window). However, since the <DELAY: n> command is a dynamic command, you should be aware of a few special considerations. See the section Dynamic Commands: Special Considerations.

<DELAY: OFF>

no abbreviation

The time delay begun with the <DELAY> command can be canceled with the <DELAY:OFF> command.

For example:

```
#0
?Click on the 'Delay' button to come out of the Delay loop.
?<CUR:3><CE><BUTTON: endloop, CREATE, "Delay", 80, 25>
?<DELAY: 5>
$<BUTTON: endloop>;<DELAY: OFF>
+Great;
#end
```

Window Activation Commands

Overview

Although the !, / and ? statement markers provide for automatic text/window direction, there are many times when you need to switch back and forth between windows from within a given statement, including the feedback portion of anticipated response statements (after the ;). In this way you will be able to increase the flexibility of your lessons. To this end there are five basic window activation commands which can be used in the *.CAL script. By using the name of the window as a command (i.e. enclosed in paired angle brackets), you activate it. After a general overview of WinCALIS windows below, the Window Activation commands will be discussed in detail.

WinCALIS makes several different types of windows available for presenting information. Some windows are always visible to students when they answer a question, other windows “pop up” and disappear (after pressing the space bar or clicking the mouse) until explicitly called for again. Some windows share the screen, while others preempt windows “beneath” them, covering windows that are otherwise visible. The different types of WinCALIS windows are listed below.

Interactive (Question) (1) This is the basic WinCALIS window. All student interaction takes place here. The Interactive (Question) Window is always visible, is re-sizable, and is shared with the Presentation (Text) Window.

Presentation (Text) (1) This window is for text which students must refer to when answering questions. It is re-sizable, and generally (though not always) shares the screen with the Interactive (Question) Window.

Tutorial (10) These are available for special supplemental information (e.g. on-line lexical or grammatical aids). They pop up only when called for (by author or student), and overwrite other information on the screen.

Popup (Directions) (1) This is a special Tutorial Window that contains the primary directions to students. It acts much like other Tutorial Windows, and can be called by pressing F9.

D (1) This is a special Diagnostic Feedback Window used for feedback to student answers. Unless explicitly directed elsewhere, all feedback will go into the D window.

WinCALIS allows authors to change the shapes and sizes of most windows in a variety of ways. WinCALIS also employs a single set of commands to control the design and appearance across all five window types. Refer to the sections Sizing Windows and Advanced Screen Design Commands.

When a window contains more information than it can contain, a scroll bar is displayed at the right-hand side of the window.

WinCALIS handles scroll bars automatically. Authors need not be concerned about the length of text vertically in a window. Students will be able to scroll through a long window with the mouse. Authors will be able to control which portion of the window is visible through the <TOPLINE> command described below. (Refer to the section Advanced Screen Design Commands). Authors should, however, be aware of the issues of screen design and presentation that arise in long windows that must be scrolled.

The section Statements and Statement Markers associated three of these windows (Question, Text, and Directions) with three WinCALIS statement markers (?, / and !). That is because these three statements send their text, by default, to their appropriate window. One might also associate the response portions of Correct/Continue, Incorrect/Remain Here, and neutral answer statements with the D window, since text on these lines is sent there. Another way to express this association is to say that these statements make their window active.

The active window is the window to which WinCALIS is currently sending text. Thus a question statement makes the Interactive (Question) Window active, and text on the line is sent to the Interactive (Question) Window. The active window can be changed at any time with the Window Activation commands described below. Consider the directions statements below. The <QUESTION> command instructs WinCALIS to make the Interactive (Question) Window active).

#0

! This goes into the Popup (Directions) Window.

!<Q> This goes into the question window.

! Where does this line go?

In the above example the <QUESTION> command redirects text to the Interactive (Question) Window. In this case, the last directions line ("Where does this line go?") goes into the Interactive (Question) Window. The exclamation point (!) designates this line as an executable WinCALIS statement that should be placed in the Popup (Directions) Window, but the <QUESTION> command

has overridden this standard. You can redirect text to the Popup (Directions) Window in this task only through use of another window-switching command, called <DIR>.

In this section, you will be introduced to the window-switching commands (or the Window “Activation” Commands) <QUESTION>, <TEXT>, <D>, <DIR>, and <F{1-10}>. In the section Window Manipulation Commands, you will be shown four basic commands for managing windows: <BLANK>, <POPUP>, and <SIZE>, as well as <IMPORT>, which is used to import text into a WinCALIS window from an outside file.

Activating the Interactive (Question) Window

<QUESTION>

Abbreviation: <Q>

You can think of the Interactive (Question) Window as the basic WinCALIS window, because the student response is always typed into it. Once the WinCALIS lesson begins, the Interactive (Question) Window is, in some sense, always in place. It may be covered up by the directions or some other Tutorial Window (refer to the section Activating the Popup (Directions) Window and the Tutorial Windows), it may coexist on the screen with the Presentation (Text) Window, but it is the place where WinCALIS goes to execute tasks. In WinCALIS's basic arrangement, the size of the Interactive (Question) Window is determined by the size of the Presentation (Text) Window: the Interactive (Question) Window consists of whatever is not taken up by the Presentation (Text) Window.

The question (?) statement puts text into the Interactive (Question) Window. The command <QUESTION> also does this. Unlike the question statement marker (?), which can only be placed at the beginning of a line, the <QUESTION> command can be placed almost anywhere in a task: in a directions(!) line in a text (/) line, in a question (?) line (this is sometimes necessary), following a separator (;) in an anticipated Correct/Continue (+) or Incorrect/Remain Here response (-), or in a spillover (&) line.

The <QUESTION> command is most useful when an author wants to set up a sequence of events in which something happens in a Text, Tutorial or Feedback Diagnostic Window, followed by a change in the Interactive (Question) Window. In the example below, the <CURSOR> command, used in conjunction with <QUESTION> and <BOLD> causes the word “white” on the first question line to be displayed in bold.

```
#question-window
?Mary had a little lamb, its fleece was white as
?snow. And everywhere that Mary went the lamb
?was sure to go.
?
?What color was the lamb?
+white;Yes, the fleece was white.
-&;<QUESTION><CURSOR:1,40><BOLD>white<BOLD>
&<D>Does this clue help?
&<QUESTION><CURSOR: 1, 40>white
```

In the feedback to the student response, the D window is the initial active window. In the above example, however, the active window switches to the Interactive (Question) Window, moves the cursor to the word white, and highlights the word. WinCALIS then switches back to the D window (with the command <D>) which gives the student the text of the clue. The D window will stay visible until the student presses the space bar or clicks the mouse. The response then makes the Interactive (Question) Window active again, and removes the highlighting from the word white.

The cursor will normally position itself at the far left of the Interactive (Question) Window, right below the last line of text. If text is added to the Interactive (Question) Window, it is added to the bottom of what had been included earlier. To automatically clear the Interactive (Question) Window after each task, the author may use the <BLANK> default option (@<BLANK>), which differs from the <BLANK> command. (See the sections “Blanking a Window” and “Task Options and Defaults”)

Activating the Presentation (Text) Window

<TEXT>

Abbreviation: <T>

As has already been pointed out, the text (/) statement puts text into the Presentation (Text) Window. The Presentation (Text) Window will normally contain examples or a reference text to which students can refer. It does not have to remain static, however. The <TEXT> command will make the Presentation (Text) Window active and redirect text there. Unlike the text statement symbol (/), which can only be placed at the beginning of a line, the <TEXT> command can be placed almost anywhere in a task: in a directions (!) line, in a question (?) line, in a text (/) line (this is sometimes necessary), following a separator (;) in an anticipated Correct/Continue (+) or Incorrect/Remain Here response (-), or in a spillover (&) line.

The Presentation (Text) Window will appear at the top of the WinCALIS screen by default. (There are commands that can put it on the bottom or the side. These will be discussed in an upcoming section, “Placing the Presentation (Text) Window.”) WinCALIS will automatically size the Presentation (Text) Window to contain the text. The remainder of the screen is taken up by the Interactive (Question) Window.

Here is an example of the <TEXT> command:

```
#text-window
/Mary had a little lamb, its fleece was _____ as snow.
?What is the missing word?
+white;<T><CUR:1,40><BO>white<BO>
&<D>Yes, the lamb was white.
-&;<T><CUR:1,49><RV>snow<RV>
&<D>What is the
&<UL>color<UL>of snow?
&<T><CUR 1, 49>snow
```

The Correct/Continue answer statement makes the Presentation (Text) Window active and fills in the blank with the Correct/Continue answer. Then the D window is made active, and the Correct/Continue answer is praised. WinCALIS switches to the Presentation (Text) Window and rewrites the word “snow” with the reverse video attribute. The D window is made active, and the text of the clue is placed into it. The D window will “pop up” and remain visible until the student presses the spacebar or clicks the mouse to proceed. The Presentation (Text) Window will again be active, and “snow” is again rewritten, this time to remove the highlight.

Activating the Diagnostic Feedback, or D, Window

<D>

no abbreviation

We have mentioned the importance of presenting appropriate feedback to student responses. This is done in comments following the separator (;) symbol in anticipated Correct/Continue, Incorrect/Remain Here, and neutral statements. However, to this point we have not dealt with how this feedback appears on the screen. By default, these comments will pop up on the WinCALIS screen in a special window, called the Diagnostic Feedback Window or D Window (which originally stood for Directed Feedback Window). The Diagnostic Feedback Window (or D Window) has several features:

- it automatically blanks itself every time new text is sent to it;
- it automatically “pops up” when text has been sent to it, waiting for the student to signal WinCALIS to continue;
- it will automatically be positioned on the screen so that its top is just below the last line of the Interactive (Question) Window; the last line of the Diagnostic Feedback Window will always be visible.
- it is otherwise unavailable to the student, after a message is read and the window closed, the student cannot get the Diagnostic Feedback Window back on screen. The D window is designed principally for short evaluation comments and hints.

The size of the Diagnostic Feedback Window is automatically calculated to be the smallest box that will contain all the text.

Feedback for student answers (text following the separator (;) in Correct/Continue, Incorrect/Remain Here, and neutral answer statements) will normally go into the Diagnostic Feedback Window. There is also a command that sends text to the Diagnostic Feedback Window from other statements, the <D> command. In the following examples of the <D> command and the <CURSOR> command (which is described in the section Positioning the Cursor and Text) are used with <BOLD> to make “Grant” on the first line of the Interactive (Question) Window appear in bold.

```
#evaluation
?Who is buried in Grant's tomb?
+Grant;Good. Grant is buried in Grant's tomb.
-Washington;<Q><CUR:1,18><BO>Grant<BOLD>
&<D>Washington was
&dead before Grant was born.
&Please try again.
```

The Correct/Continue answer (+) statement puts the author's comment into the Diagnostic Feedback Window. Students who type in “Grant” will see the Diagnostic Feedback Window at the bottom of the screen with the message, “Good. Grant is buried in Grant's tomb.” The Diagnostic Feedback Window will remain until the student has finished reading it, then WinCALIS will continue with the next task. Notice that the author must specify the <D> command, because the Interactive (Question) Window had been made the active window in the previous line with a <QUESTION> command.

Activating the Popup (Directions) Window and the Tutorial Windows

```
<DIR>
no abbreviation
```

```
<F{1-10}>
no abbreviation
```

The Popup (Directions) Window is the window students access via the [F9] key or by selecting it from the Options menu. You have seen how text can be entered into the Popup (Directions) Window with a directions statement. You may also make the Popup (Directions) Window active by using the <DIR> command. As with other window-switching commands, <DIR> may appear in directions, text, question, and spillover statements, as well as in the response portion of Correct/Continue or Incorrect/Remain Here answer statements.

The author may provide the student with information other than that provided by the Popup (Directions) Window. This is done through the use of ten Tutorial Windows. The [1-10] parameter specifies the number of the Tutorial Window to be displayed. There are no one-character line symbols which direct text to the Tutorial Windows. They must be made active with one of the ten Tutorial Window commands, which are:

<F1> <F2> <F3> <F4> <F5> <F6> <F7> <F8> <F9> <F10>

These correspond to the ten basic function keys on all PC keyboards (or, more precisely, to the ten function keys used in conjunction with the Shift key).

Text following the <F{1-10}> command will be placed in the specified Tutorial Window. Students can “pop up” the Tutorial Windows for reference by holding down the Shift key while typing the appropriate function key simultaneously, or by selecting the desired tutorial from the Options menu. For example, pressing Shift+F7 will pop up the F7 Tutorial Window. the following is an example of the <DIR> and <F#> commands:

```
#new-dir
!<F1> This is new text for the F1 window.
!<F2> This is new text for the F2 window.
!<DIR> These are new directions.
```

Here new text is placed in the F1 and F2 Tutorial Windows. The Popup (Directions) Window again becomes the active window with the <DIR> command, and the new directions information is added. The student can pop up the text in the F1 window by pressing Shift+F1 on the keyboard. Similarly, the contents of the F2 window are accessible by pressing the key combination Shift+F2.

Window Manipulation Commands

Blanking a Window

<BLANK [W]>

Abbreviation: <BLA [W]>

Before discussing the other window-related commands, we will introduce a basic window command, <BLANK>, which can be used to clear away unwanted text from your screen. The Feedback Diagnostic Window, will automatically blank, erase its entire screen of old text, when new text is entered into it. In other windows, texts builds as it is added. The command <BLANK> will erase all text from a window. Everything written in a window, both text that is visible and text that may not be currently visible, is erased. The cursor is reset to the top row and first column. The <BLANK> command can be used in two ways:

```
#blank
?<BLA>
?<BLA F1>
```

The first question (?) statement in the previous example blanks the active window (the window currently receiving text, in this case the Interactive (Question) Window). The second question line blanks a window that is not currently active (in this case the F1 Tutorial Window). The <BLANK> command allows the author to get a “fresh start” with the screen.

"Popping Up" the Directions and Tutorial Windows

<POPUP DIR>

<POPUP F{1-10}>

Abbreviation: <POP [W]>

In the previous section it was shown that the Directions and Tutorial Windows could be accessed through the function keys: the Popup (Directions) Window through pressing the F9 key, the ten tutorial windows through simultaneously pressing Shift and the appropriate function key. Another way to make these windows available is with the <POPUP> command. <POPUP> will cause a window to pop up onto the screen. A student will read the relevant information from the window, and press the spacebar to continue. The window will appear in a locked state to the screen: the window will have to be closed before the student can continue. <POPUP> will pop up a window that you have specified. For example:

```
#third-person-pronoun
!<F1>
!   Pronouns:
!           Singular      Plural
!   1st      I            we
!   2nd      you          you
!   3rd      he, she, it   they
!<POPUP F1>
?Supply the correct form of the pronoun indicated.
?She gave _____ the book. (he)
+him;Yes. "She gave him the book" is correct.
-he|his;<POPUP F1>
```

In the above example, a chart of pronouns is placed into the F1 window. This chart is shown to students before the question. A student who incorrectly answers “he” or “his” is shown the chart again. The Popup (Directions) Window and the Tutorial Windows can also be popped up in a no-lock state, i.e. they can be resized and stay on-screen and the student can continue answering the questions.

<POPUP DIR [NOLOCK] >

<POPUP F{1-10} [NOLOCK]>

Abbreviation: <POP F{1-10} [NOLOCK]>

```
#third-person-pronoun
!<F1>
!   Pronouns:
!           Singular      Plural
!   1st      I            we
!   2nd      you          you
!   3rd      he,she,it    they
!<POPUP F1 NOLOCK>
```

?Supply the correct form of the pronoun indicated.
 ?She gave _____ the book. (he)
 +him;Yes. "She gave him the book" is correct.
 -he|his;<POPUP F1 NOLOCK>

In the above example, the Tutorial Window will pop up twice in a no-lock state.

Placing the Presentation (Text) Window

<SIZE TEXT: [HIGH|LOW]>

Abbreviation: <T>

An author has the option of placing the Presentation (Text) Window either at the top part of the screen or at the bottom. One simple command at the beginning of the Presentation (Text) Window will tell WinCALIS where to put the text. The default setting is the Presentation (Text) Window at the top of the screen. To place it at the bottom of the screen, use <SIZE T: LOW>. To place it again at the top of the screen, use <SIZE T: HIGH>.

Sizing Windows: the <SIZE> Command

An author may wish to change the default window sizes set by WinCALIS. In order to do so, the <SIZE [W]:top,left,bottom,right> command must be used. The syntax for this command is:

<SIZE [W]:top,left,bottom,right>

no abbreviation

Window:	Syntax:	Abbreviation
Popup (Directions)	<DIR>	none
Interactive (Question)	<QUESTION>	<Q>
Presentation (Text)	<TEXT>	<T>
Tutorial	<F[1-10]>	none
Diagnostic Feedback	<D>	none

Window Dimensions (top, left, bottom, right):

top	any number between 0 and 100
left	any number between 0 and 100
bottom	any number between 0 and 100
right	any number between 0 and 100

In order to understand how to use the <SIZE [W]:top,left,bottom,right> command you should be aware of the relation between the Size/Grid System and the <SIZE [W]:top,left,bottom,right> command, as well as the difference between the screen and a window. The screen is what you might call the total of the display capacity of your monitor. A window is a rectangular area on your screen in which information is displayed. The screen as employed by WinCALIS, may be thought of as containing a grid of rows and columns from 0 to 100 for length and width. The grids are absolute to the screen itself and not to the particular window.

To size a window and locate it in the middle of the screen you enter the following parameter into the <SIZE> command:

<SIZE [W]:33,33,66,66>

The first parameter, “33” informs WinCALIS that the top of the window should be located 33% down from the top of the screen. The second “33” causes the left border of the window to appear 33% of the screen from the left-hand side of the screen. The first “66” parameter tells WinCALIS to locate the bottom border of the window 66% down from the top of the screen. The last parameter causes the right-hand side of the window to appear 66% from the left-hand side of the screen.

The [W] option again offers the opportunity to define the window the sizing should be applied to. If not specified, WinCALIS will apply the sizing to the window that you are currently editing in WinCALIS Author, for example, to the Popup (Directions) Window when you enter the <SIZE: top,left,bottom,right> in the Popup (Directions) Window. For example:

```
#window-size
/<BLA><SIZE 0,0,50,50><SET BC CYAN>
/<CUR:4,> <CE>This Text Window takes up
/<CE>one half of the total screen.
?<BLA>
?<SET FS MEDIUM><CE>Which corner is taken
?by the Presentation (Text) Window?
?
?<TAB:4>(a) upper left
?<TAB:4>(b) upper right
?<TAB:4>(c) lower right
?<TAB:4>(d) lower left
?
?<CE><M:prompt> <CUR:prompt>
+a;<CE>Correct
-b;<SIZE T 0,50,50,100><CE>Incorrect, this is the
&upper right corner
-c;<SIZE T 50,50,100,100><CE>Incorrect, this is the
&lower right corner
-d;<SIZE T 50,0,100,50><CE>Incorrect, this is the
&lower right corner
```

Remember that when you size the Presentation (Text) Window, you may also have to size the Interactive (Question) Window

In the above example, the Presentation (Text) Window initially takes up the upper left corner of the screen. Pressing “b” as a student response will cause WinCALIS to display the Presentation (Text) Window at the upper right corner, while the message “Incorrect, this is the upper right corner” is displayed in the D Box. After pressing the spacebar, the Presentation (Text) Window is displayed in the upper left corner again. This occurs because WinCALIS repeats the entire task (including the Presentation (Text) Window lines). The <SIZE: top,left,bottom,right> command remains active until another <SIZE: top,left,bottom,right> command is encountered in the script.

Importing Other Files into WinCALIS

You have seen how the various window-switching commands allow the author to direct text to various WinCALIS windows. Up to this point, however, all text that went into these windows had to be present in the WinCALIS script. There are times when an author will want to use the same text for various lessons, as is often the case with basic tutorial screens. WinCALIS provides a way to import other DOS text files into the lesson. This is done with <IMPORT> command. The basic structure of this command is:

```
<IMPORT [W] filename.ext>
```

Abbreviation: <IMP [W] filename.ext>

The filename.ext is the name of the imported file and [W] is the window to which the file is being imported. (If no window is specified, the file is imported to the current window.) The file name may have any legal DOS name. WinCALIS will search for the file in the subdirectory where it found the *.CAL script file. The imported file must have been saved as a Unicode file. Such a file can easily be produced when using the WinCALIS Author Scratchpad window with the Document file: Save as.. option. The following is an example of <IMPORT>:

```
#0
!Answer the questions as best as
!you can.
!
!At any time you may press
! [Shift+F1] to receive general help on WinCALIS
! [Shift+F2] to look at the WinCALIS keyboard
! [Shift+F3] to review the modal verbs
!<F1><IMPORT: help.bsc><IMPORT F2: keyboard>
!<IMPORT F3: modals.gra>
```

This example sets up several Tutorial Windows for student reference, general help is available in the F1 window, a keyboard diagram is in the F2 window, and a chart of modal verbs is imported into the F3 window. Each of these windows may be relevant to many WinCALIS exercises. By placing each in a separate file and importing them as needed, an author need not reproduce the same information in every script.

Files that are imported with the <IMPORT> command may include WinCALIS commands with the following exceptions and restrictions:

- <IMPORT> and <POPUP> cannot be included in imported files under any circumstances.
- Commands which redirect WinCALIS to a particular window, including <DIR>, <Q>, <T>, <F1> and <D>, must be excluded from imported files.
- Commands which can be directed to a window other than the one in which they appear including <BLANK>, <CURSOR>, <CLEAR>, <MARK>, <SIZE> and <TOPLINE>, must not contain these designations in an imported file. The command must be directed to the current window (the window into which the file is being imported). For example, <CURSOR: 0,0> is a legal command for an imported file, while <CURSOR DIR: 0,0> is not.

In general, include commands in an imported file that are designed to work only within the window to which the file is being imported.

Like WinCALIS scripts, files brought in with the <IMPORT> command must be in Unicode format. Unlike WinCALIS scripts, they are not themselves executed by the WinCALIS program. They are brought in for viewing by the student. Therefore, do not put WinCALIS statement symbols (@, #, ?, etc.) at the beginning of the lines in imported scripts unless you want them to show up on the WinCALIS screen. Also, do not give your import files names with the extension .CAL or they will be displayed in your students' WinCALIS menu.

Naming Tutorial Files with the <TUTNAME> command

<TUTNAME F{1-10} description>

Abbreviation: <TN F{1-10} description>

Tutorial Window Options: F{1-10} any Tutorial Window from F1 through F10.

Description: your description for the Tutorial Window, not to exceed 50 characters.

In using Tutorial Windows, an author will soon notice that often a descriptive name for Tutorial Windows such as “Irregular Verbs” or “Glossary of Computer Terms” is much more clear than “Tutorial 1” or “Tutorial 2,” which is displayed as a default by WinCALIS. Therefore you will find it convenient to name your Tutorial Windows by using the <TUTNAME> command.

Note: If you specify no Tutorial Window (F{1-10}), WinCALIS will assume that the name that you are defining is meant for the currently activated window. This implies that the <TUTNAME description> must be preceded in this case by a command that activates the Tutorial Window in question, i.e. <F{1-10}>.

```
#sample_tutname
!In this lesson you will be asked to answer the
!questions about the text. If you like you can
!consult the glossary of
!computer terms during
!the lesson by pressing Shift+F1.
!<F1><TN Glossary of Computer Terms>
!Glossary of computer terms
!RAM: acronym for 'random access memory';
!semiconductor-based memory that can be read or
!written by the microprocessor or other hardware
!devices.
!ROM: acronym for 'read only memory';
!semiconductor-based memory that can contains
!instructions or data that can be read but not
!modified.
!<TN F2 General Information><IM F2:help>
```

Tutorial Windows F1 and F2 will be displayed with the labels “Glossary of Computer Terms” and “General information” respectively. When the student pulls down the Tutorials menu option this text will also appear next to F1 and F2.

Note: In the case of the F2 Tutorial Window, the <TUTNAME> command must precede the <IMPORT> command. Any text or command after the <IMPORT> command will be ignored completely by WinCALIS.

If the <TUTNAME> command is used within the text of an activated tutorial window, as is the case for the F1 Tutorial Window in the above example, the author does not have to specify which window is being named; the current one is assumed. Otherwise, if the <TUTNAME> command is used outside of the tutorial window, as shown in the case of the F2 Tutorial Window in the example, the author must specify which tutorial window is being named or the command will be ignored by WinCALIS.

Advanced Screen Design Commands

Screen design refers to the creation of informative and effective lessons. Screen design issues include using space and highlights to emphasize important information, structuring text to convey relationships, and allowing students to manipulate text directly to observe cause and effect. WinCALIS provides several commands to help authors design effective screens. Some of the basic ones (like <BOLD>, <CENTER>, <TAB: n> and others) have already been discussed in a previous section. This section will explain ten of the more advanced commands:

<CURSOR>	<MARK>	<CLEAR>	<FILL>	<SET>
<LAST>	<N>	<CORRECT>	<TOPLINE>	<BEEP>

If you have worked with graph paper or are familiar with grids, the following should be fairly straightforward. If this method of screen design is entirely new to you, experiment. Nothing can replace trial and error!

Positioning the Cursor and Text

Placing text at particular places on the screen is done in WinCALIS by means of the <CURSOR: row, column> command.

<CURSOR: row, column>

Abbreviation: <CUR: row, column>

Parameters:

- row- the absolute row number in the window
- column- the absolute column number in the window.

The most fundamental concept for WinCALIS screen design is that of the cursor location. Most word processors employ a flashing cursor that invites users to type a character. The location of the flashing cursor indicates where the next letter will appear on the screen. As a user types more characters, the cursor moves to the right and down the screen.

WinCALIS also uses a flashing cursor, but only in one window (the Interactive (Question) Window), and only when the user is being asked to type an answer. Every window, however, has a cursor location, the row and column within the window where the next character will appear. This location also moves to the right and down as text is added to the window. It might be useful to consider that every WinCALIS window maintains a cursor, which differs from the flashing cursor used by word processors only in that WinCALIS cursors are usually invisible.

Most word processors use arrow keys to move the cursor up or down, left or right, without changing the appearance of text on the screen. WinCALIS also allows authors to change the location of the cursor.

The command <CURSOR: row, column> changes the location of the cursor, and, thus, the location within a window where the next character will appear. <CURSOR: row, column> moves the cursor location to the specified row and column.

The example below shows the use of the <CURSOR: row, column> command to redirect the cursor both in the Interactive (Question) Window (in a line starting with ?) and in the Presentation (Text) Window (following the <TEXT> command).

```
#snow
/<CUR:1,1>Mary had a little lamb, its fleece was
/white as snow. And everywhere that Mary went,
/the lamb was sure to go.
?What color was the snow?
?  >
?<CUR: 2,10>
+white;Very good
-red;<T><CUR: 1,40><UL>white
&<D> Come, come. You can do better!
```


The third question (?) line in the example above moves the cursor back up to the line with the > prompt. Students will then type their answer on that line. The Incorrect/Remain Here answer line makes the Presentation (Text) Window active, and moves the cursor to the beginning of the word “white.” The command overwrites the word, placing an underline beneath it (white).

Abbreviation:

<CUR [W]: row, column>
<CUR [W]: +row, -column>

The basic outline of the cursor positioning command is:

<CURSOR [W]: row, column>
<CURSOR [W]: +row, -column>

where the row is a number that indicates the row number of the specified window ([W]), and column is a number that indicates the column number of the specified window. You may think of each window as having its own coordinate system, independent of what the student sees on the screen. Therefore, in WinCALIS default behavior, row 1 of the Interactive (Question) Window may appear in the middle of the WinCALIS screen, as the first line below the Presentation (Text) Window box. Row 1 of the Interactive (Question) Window may be put at any number of positions on the WinCALIS screen.

The basic <CURSOR [W]: row, column> command requires a row and column in which to place the cursor. The active window (the window currently receiving text) is the default, unless a different window is specified (indicated by the W option above). Valid windows which may be specified in a <CURSOR> command are the Text, Question, Directions, Tutorial (F1 through F10), and Feedback Diagnostic Window. The row and column specified in the <CURSOR [W]: row, column> command may be absolute positions, relative positions, or default. The first cursor command outlined above shows the form for writing absolute positions. Relative cursor position commands, which contain numbers preceded by a plus sign (+) or minus sign (-), start at the current cursor position and move down or up (in the case of row), or right or left (in the case of column) by the number of rows/columns that are specified.

If row or column numbers are missing, WinCALIS uses the current row or column.

<CUR F1: 5> Move cursor to row 5 in the current column of the F1 window

<CUR F1: ,5> Move cursor to column 5 in the current row of the F1 window

Note the following examples.

```
#cursor
?Fill in the blank
?Mary had a little ____.<CUR: 2, 18>
+lamb;Yes. “Mary had a little lamb” is correct.
&<CUR Q: 2, 18><Q><BO>lamb
```

In the previous example, WinCALIS will move the cursor to row 2, column 18 of the active window. Since it occurs in a question statement and there has been no switching of windows, the active window is the Interactive (Question) Window. Row 2, column 18 is the beginning of the four blanks, which is the location students will now type their answers. The correct answer response gives the student positive feedback, then moves the Interactive (Question) Window cursor to the beginning of what is now the student's answer, where the word “lamb” is rewritten and highlighted.

```
#nursery-rhyme
?Fill in the missing word:
```

?
 ?Mary had a little lamb,
 ?its fleece was _____ as snow.
 ?And everywhere that Mary went
 ?the lamb was sure to go.<CUR: -2, 16>
 +white;<Q><CUR:, 16><BOLD> white <BO> as
 &snow.
 &<D>Yes, its fleece was white as snow.

The final command in the question statement moves the cursor up two rows, and to absolute column 16 (the beginning of the blanks). The correct answer statement leaves the cursor on the same row, moves back to column 16 (which should be the beginning of the student's answer), and rewrites the word "white" in bold letters. Relative positions in the <CURSOR> command are particularly useful if the interaction of several tasks makes it difficult to predict what the absolute row should be. Note that, when counting columns, the WinCALIS statement symbols (!, ?, /, +, -) do not count as one column. You may wish to think of it as column number 0. Similarly, any WinCALIS character which does not show up on the screen (such as commands or other features enclosed within angle brackets) should not be counted when determining columns.

The <CURSOR> command may be used to move the cursor to a portion of the window not previously visible. If the row is the next row up or the next row down, the window slides, or scrolls one line. If it is more than one line above the top or below the bottom of the visible screen, the visible portion of the window is centered so that the new cursor row appears in the middle of the window.

Positioning the Cursor Without Specifying Coordinates

<MARK: marklabel>

Abbreviation: <M: marklabel>

definition: the name given as marklabel

The <MARK:marklabel> command allows an author to direct a cursor to a point in a window without specifying the actual numerical values of the coordinates (the row and column). The author first "marks" a point in the window with the <MARK:marklabel> command, giving the point a unique label.

This point can be accessed with the <CURSOR> and <TOPLINE>. Note that a mark has both a row and a cursor position associated with it. Therefore, a reference to one mark will suffice in a <CURSOR> or <TOPLINE> command. WinCALIS makes no distinction between the use of upper and lower case in label names. In other words, <MARK:LABEL>, <MARK: Label>, and <MARK: label> are all considered the same.

Marks are window-specific: A mark created in a window can only be accessed in that window. (This means that different marks with the same name can exist in different windows.) If a <CURSOR> or <TOPLINE> command is directed to a mark which doesn't (or does not yet) exist, the command is ignored. Marks should not be given labels which conflict with the three commands that access them. This means that these labels:

- should not start with a digit (0 through 9)
- should not start with a plus sign (+)
- should not start with a minus sign (-)
- should not be the same as a window name, i.e., "q" or "question"; "t" or "text"; "dir" or "direction"; "d"; or the letter "f" followed by a digit.

```
/Here's an easy one: <M: T1>
?Who's buried in Grant's tomb?
+Grant;Very Good!
-&;<T><CUR: T1>Easy for some, I guess.
```

Used with the <T> command <MARK> is used to alter the Presentation (Text) Window after a wrong answer. For the <CURSOR> and <TOPLINE> commands <MARK> is in almost all circumstances easier to use than numerical values.

Clearing Parts of a Window

<CLEAR [W] [LINE|LINE LEFT]>

Abbreviation:

```
<CLE [W] LINE>
<CLE [W] LINE LEFT>
```

Two basic window commands, <BLANK> and <CLEAR>, can be used to clear away unwanted text from your screen. The <BLANK> command, as is explained in “Blanking a Window,” erases an entire window. The <CLEAR> command can be used to erase the screen from the point at which it is inserted (to the right or the left), or to erase a single line. <CLEAR> is often used in conjunction with the <MARK> command.

Note: The [W] shown in the abbreviations for the CLEAR commands is replaced by the name of a window (optional). For example, to clear text to the left of the cursor in the Interactive (Question) Window from the Feedback Window you would use the command <CLEAR Q LINE LEFT>

```
#clear_sample
/Mary had a little lamb,<M T1>Its fleece was white...
/Everywhere that Mary went the lamb was sure to go
```

```
#1
?What did Mary have?
+lamb
```

```
#2
?<T><CUR: T1><CLEAR: LINE>
?What color was the fleece?
```

In the above example, for task #2, only the text portion to the right of the T1 mark is deleted in the first line.

There are three versions of the <CLEAR> command:

- <CLEAR> clears text from the cursor position down to the bottom of the window
- <CLEAR: LINE> clears text from the cursor position right to the end of the line
- <CLEAR: LINE LEFT> clears text from the cursor position left to the beginning of the line

It is also possible to use the <CLEAR> command in combination with the <CURSOR> command and a specified set of coordinates as explained in the section Positioning the Cursor and Text.

Generating and Sizing Blanks

<FILL [length] [#task] [character]>

no abbreviation

Parameters:

length- any absolute number to indicate the number of spaces or characters you want to print

#task- the task label of the task containing the longest Correct/Continue Anticipated Response which will be taken as the number of characters/spaces to print; the actual task label must be preceded by the task label statement marker

#character- the character you want to fill the gap with

This command is especially useful as an automatic blank generator. When the command <FILL> is encountered, WinCALIS will put underline characters () on the screen. When the command is used in its most basic form (<FILL>) it puts up the same number of characters as the longest anticipated correct response in that task. (Refer to the discussion of the <LENGTH> option in the section Task Options and Defaults.)

Other options:

- Put in a number to print out a specific number of characters.
Example: <FILL 5> will print out five underline characters.
- Put in the label of another task (following a # character) to create a blank based on the longest anticipated response in that task.
Example: <FILL #12 *> will create a blank made up of asterisks (*) and equaling the longest Correct/Continue Anticipated Response of task #12.

Some restrictions:

- Do not use a digit (0-9) or a so-called "control" characters (those which are not included on a standard keyboard).
- In order to use one of the following characters: , ; : \ (space) as the fill character, precede it with the backslash (\) character.

Example: <FILL \,> will create a blank made up of commas. The character > cannot be used as a fill character.

Note: It is possible to set the number of characters printed to a constant length with the <SET> command. (See the section "Defining Certain Numeric Values").

Defining Certain Numeric Values

<SET: parameter n|OFF>

<SET: [DIR|F{1-10}] [ENABLE|DISABLE] [SCROLL ON|OFF]>

no abbreviation

The <SET> command gives WinCALIS certain numeric values that remain in place until changed or removed with another <SET> command. The parameter value is limited to integers between 1 and 99. Currently, the possibilities include:

- <SET: ANSWER *n*>, where *n* is the number of characters that a student can type for a particular answer. It is only in effect when the <LENGTH> task option is in effect. See the section Task Options and Defaults.
Example: <SET: ANSWER 8> will limit the student to eight characters for his response.
- <SET: CORRECT *n*>, where *n* is the number of characters, including trailing blanks, to be printed with the <CORRECT> command.
- <SET: FILL *n*>, where *n* is the number of characters, including trailing blanks, to be printed with the <FILL> command.

Note: This can also be set within each <FILL> command. See the section on Generating and Sizing Blanks.

- <SET: LAST *n*>, where *n* is the number of characters, including trailing blanks, to be printed with the <LAST> command.

Note: this can also be set within each <LAST> command. See the section Printing the Student's Last Response to the Screen.

Use the word “OFF” to return any of the above options to the WinCALIS default behavior. For example, if the <LENGTH> task option is in effect and you use the command <SET: ANSWER 10>, the student can type up to ten characters on the screen. The command <SET: ANSWER OFF>, resets the answer to the default limits set by the longest anticipated Correct/Continue response. The <SET> command can also be used to (partly) disable various Tutorial Windows.

- <SET: [F1] [F2] [F3] ...[F10] DISABLE> will disable the listed Tutorial Windows.
- <SET: [F1] [F2] [F3] ...[F10] ENABLE> will enable the listed Tutorial Windows to be displayed when the Shift + function key combination is pressed.
- <SET: [F1] [F2] [F3] ...[F10] SCROLL OFF> will disable scrolling within the listed Tutorial Windows.
- <SET: [F1] [F2] [F3] ...[F10] SCROLL ON> will enable scrolling within listed Tutorial Windows.

Combinations of the above commands are also possible.

The command <SET: [F3] [F5] ENABLE SCROLL OFF> will allow the student to access the Tutorial Windows, by pressing Shift+F3 or Shift+F5 but prohibit scrolling within these Tutorial Windows.

The Tutorial Windows are available by default and can be scrolled if they are defined by the author in the script or using a command such as

<F1><IMPORT: filename.ext>

where a file such as “filename.ext” exists. Therefore, the <SET: ENABLE> and <SET: SCROLL OFF> commands are only used if a window has been disabled or scrolling has been turned off. For a review of the <IMPORT: filename.ext> command refer to the section Importing Other Files into WinCALIS.

Displaying the Student's LAST Response

<LAST [length]>

no abbreviation

Parameter:

length- a numeric value defining the number of characters of the last student response to be displayed on screen.

This command will print to the screen the last response entered by the student. This can be used to keep student's previous responses on the screen as a reference. For example:

```
#1
? . . .
?
.
.
-&; No, that's wrong.<F3><LAST><CUR: +1,1>
```

In the above example, the student's wrong answers are shown in the F3 Tutorial Window. You can put a number in the <LAST> command to format it to a certain length: The response will fill up exactly that number of spaces on the screen. It will truncate a response that is too long, and pad a shorter response with blanks. For example, the command <LAST: 10> will print out "Christophe" (the first ten characters) for the student response "Christopher," and will print out "Columbus " (the full word followed by two spaces) for the eight-letter response "Columbus." This number can also be set with the <SET> command (See the Section "Defining Certain Numeric Values" above.)

Printing the Correct/Continue Answer as it Appears in the Script

<CORRECT [length] [#task]>

Abbreviation: <COR [length] [#task]>

Parameters:

length- a numeric value defining the number of characters to be displayed during the first Correct/Continue Anticipated Response in the script.

#task- the task label of the task referred to when displaying an Correct/Continue Anticipated Response

```
#1
/ This response is <M: T1>
? . . .
.
.
+right; This is the correct response.
&<T><CUR: T1><RV><CORRECT><RV>.
.
.
-<GIVEUP>;<T><CUR: T1><RV><CORRECT><RV>
```

In the above example, the Correct/Continue answer (right) is printed in reverse video to the Presentation (Text) Window at mark T1 if the student answers correctly or gives up. WinCALIS will print out the Correct/Continue response exactly as it appears in the script. Therefore, to avoid student confusion do not use wild cards in the first anticipated Correct/Continue response.

One can put a number in the <CORRECT> command to format it to a certain length: the response will fill up exactly that number of spaces on the screen. It will truncate a response that is too long, and pad a shorter response with blanks. For example, the command <CORRECT: 10> will print out "Christophe" (the first ten characters) for the Correct/Continue answer "Christopher," and will print out "Columbus " (the full word followed by two spaces) for the eight-letter Correct/Continue answer "Columbus." This number can also be set with the <SET> command (See the section "Defining Certain Numeric Values.") The task to be referred to when selecting an Correct/Continue Anticipated Response can also be defined. <CORRECT: 5 #4> will display the 5 first characters of the first Correct/Continue Anticipated Response of task 4 in the script.

Printing the Sequence Number of the Current Task

<N>

no abbreviation

The <N> command will print to the screen at the location of the command the number of the current task in its sequence in the task list, as encountered by WinCALIS during execution of the script.

If you desire to have item numbers appear on the screen, using the <N> command saves you from having to keep track of task numbers or renumber tasks by hand, if you insert or delete tasks in the task list.

Changing the Top Line Visible on the Screen

<TOPLINE [W]: [row|mark]>

Abbreviation: <TOP [W] [row|mark]>

Syntax:

row - the absolute row number of a particular window

(or)

mark - a mark specified in a particular window

Window:	Syntax:	Abbreviation
Popup (Directions)	<DIR>	none
Interactive (Question)	<QUESTION>	<Q>
Presentation (Text)	<TEXT>	<T>
Tutorial	<F[1-10]>	none
Diagnostic Feedback	<D>	none

The <CUR: row, column> command operates independently of your window's appearance on the screen. A command which is helpful in setting up the position of your text on the screen is the <TOPLINE:[W] [row|mark]> command, which can change the portion of a window which is visible on the screen. The <TOPLINE: [W] [row|mark]> command makes the specified row the first line at the top of the window box. For example:

```
#pronoun-tutorial
```

```
!<F1><IMP: pronouns>
```

```
?For help with pronouns type one of the following:
```

```
?
```

```
? 1 - First person pronouns
```

```
? 2 - Second person pronouns
? 3 - Third person pronouns
? Q - Done
+Q&/q&
1;<TOP F1: 1><POP F1>
2;<TOP F1: 25><POP F1>
3;<TOP F1: 50><POP F1>
```

In this example a Pronoun Tutorial file (pronouns) has been imported into the F1 Window. This tutorial contains three screens of information (lines 1-16 are about first person pronouns, 25-40 are about second person pronouns, 50-65 are about third person pronouns). Entering 1, 2, or 3 causes the appropriate portion of the F1 Window to be made visible by changing which line will appear at the top of the window.

The <TOPLINE [W] [row/mark]> command works with the Question, Text, Directions, and F{1-10} Tutorial Windows. There are some limitations on the use of <TOPLINE> in the Interactive (Question) Window that relate to cursor positioning. WinCALIS will ignore a <TOPLINE [W] row/mark> command in the Interactive (Question) Window if the current cursor position in the Interactive (Question) Window will be pushed off the screen through repositioning the top line. (The cursor in the Interactive (Question) Window must always be visible to the student.)

Note: The <TOPLINE: [W] row/mark> command can also be used with a mark instead of a row number. In that case WinCALIS will use the line in which the mark occurs, as the top line of the window. For example, <TOPLINE T: mark1> will display the line in which the mark <M:mark1> occurs as the top line for the window.

Sounding a Beep: the BEEP Command

<BEEP [{2-10}]>

no abbreviation

Option ({2-10}): a number from 2 through 10 to indicate the number of beeps.

This command will insert a beeping noise. You can specify a number from 2 to 10 to make WinCALIS beep that number of times.

```
#1
? . . .
+hello; This will beep once.<BEEP>
+goodbye; This will beep three times.<BEEP 3>
```

<BEEP> is a dynamic command, and will not sound until the window it is inserted in has been viewed by the student. (See the section Dynamic Commands: Special Considerations).

In WinCALIS, certain dynamic commands, if directed to a hidden Tutorial Window, are put on hold until that window is displayed. This adds power to WinCALIS, but does require special consideration by authors to avoid unexpected occurrences.

Media Commands

WinCALIS incorporates Media Commands for installation of graphics, audio and/or video clips in WinCALIS lessons. It is even possible to start another Windows application from within WinCALIS. These commands 'import' media materials in WinCALIS. They are quite static when used in the regular WinCALIS windows (Presentation (Text) Window, Interactive (Question) Window). However, you should remember they can also be used in the feedback portion of WinCALIS scripts. As in other multimedia software, WinCALIS allows interactivity through the use of media commands. In this

section there is an overview of the media commands. For further information, refer to the section Multimedia in WinCALIS, which discusses the WinCALIS Multimedia Editor, and to the Multimedia Editor On-line Help itself.

Note: Since CALIScript is extended often, not all commands are covered in this section. Please refer to the section Multimedia in WinCALIS or CALIScript Quick Reference Guide for an exhaustive survey of all media commands, and/or the appropriate “readme.txt” files.

Incorporating Graphics

<GRAPHIC:filename.bmp [name]>

Abbreviation: <GR [filename.bmp] [name]>

Syntax:

filename- the filename given to the *.BMP file. The extension “.BMP” may be included, but is not obligatory.

name- optional “handle” or name by which you refer to the graphic with the <HOTSPOT> special response.

The <GRAPHIC> command is used to import graphics files into WinCALIS. Consider the following example:

```
#sample_graph1
/Look at this map and answer the following questions.
/<GR: map.bmp>
```

In the above example, the image “map.bmp” will be displayed in the Presentation (Text) Window. WinCALIS supports Device Independent Bitmaps (DIB) with the extension .BMP. A DIB is a standard graphics file format supported by Windows 3.1 and higher and WinCALIS. “Device Independent” means that a graphic can be displayed regardless of the graphics card installed on the computer the student is using. Graphics created using 256 colors may not look as good on a monitor capable of displaying only 16 colors as graphics created using just 16 colors to begin with.

The Windows 3.1 graphics editor Paintbrush, and Windows 95 editor Paint, stores bitmapped files in this format. No conversion of these files is necessary.

The upper left-hand corner of the graphics file will appear at the point that the <GRAPHIC> command is inserted.

In order to identify a hotspot on a graphic, you need to give the graphic a “handle” or name by which you refer to it in the HOTSPOT special response. The name can be simply the 1-8 letter filename for the *.BMP file, or any other string of letters you choose.

In the sample script below, the graphic “surfbrd.bmp” is given the name “surfbrd” with the command

<GRAPHIC: surfbrd.bmp surfbrd>

```
#sample_graph2
?Click the image if you want to get more information.
?<GRAPHIC:surfbrd.bmp surfbrd>
?
?
?
?
```

?

...

=<HOTSPOT surfbrd>;Surfing is a water sport that
&involves planing in a controlled fashion on the
&sloping portion of a wave as it moves toward
&shore.

In the above example clicking the image with the mouse will cause WinCALIS to display the D Window with some information on surfing.

See Also: Making Hotspots in Graphics
More Detailed Information on Graphics

Tip: Graphics with a lot of unneeded white space around them waste disk space and take longer to display. Keep this in mind when creating graphics. In Windows Paintbrush, you can save a graphic without unnecessary white space by first using the “Pick” tool to mark a box around just the portion of your window you need, then in the Edit menu, use the “Copy to...” command to save the picked portion to a file, rather than saving the whole window with the File menu “Save” command.

Incorporating Audio and Video

Note: For more information on using WinCALIS Author to insert the commands to control modes and location of video, refer to the section Multimedia in WinCALIS or CALIScript Quick Reference Guide.

<AUDIO: XXX>

Abbreviation: <AUD>

The <AUDIO:XXX> command can have many variations of which the <AUDIO: clipname> is actually the most basic form. It will cause WinCALIS to play the specified clip as it was defined using the Multimedia Editor.

Just as your audio equipment allows more options than just “playing,” WinCALIS allows you to pause or resume the audio file <AUDIO:PAUSE>; <AUDIO:RESUME>, to choose the left, right, or both audio channels; <AUDIO: LEFT>; <AUDIO: RIGHT>; <AUDIO:STEREO> and to play, record, and delete digital audio wave files <AUDIO:PLAYfilename.wav>; <AUDIO:RECORD filename.wave.millisecs>; <AUDIO:DELETE filename.wave>.

<VIDEO PLAY clipname>

Abbreviation: <VID>

Syntax:

clipname- the name given to the clip when using the Multimedia Editor to define video clips

#sample-video

?watch this!

?<VIDEO:PLAY shot1>

+all right!;

In the above sample script, a video scene named “shot1” from a digital video file, a laserdisc, or a video cassette will be shown to the student.

WinCALIS supports a number of options for video display:

<VIDEO:SIZE[FULL|QUARTER|EIGHTH|SIXTEENTH]>

This command allows you to specify the size of the video image. The size of a video image could appear as one eighth of the screen, one fourth of the screen or the full screen. If you do not specify an image size the default is <VIDEO: SIZE QUARTER>

<VIDEO:LOC [TL|TR|TC|CE|LL|LC|LR]>

This command allows you to specify where on the screen the video image should appear. It allows the following options:

TL = Top left portion of the screen.
TC = Top center portion of the screen
TR = Top right portion of the screen
CE = Center portion of the screen
LL = Lower left portion of the screen
LC = Lower center portion of the screen
LR = Lower right portion of the screen

SPELLMARKing the Student Answer

<SPELLMARK>

Abbreviation: <SPMK>, <SPCK>

Synonym: <SPELLCHECK>

WinCALIS has a unique way of checking spelling mistakes. Unlike a word processor, WinCALIS does not employ a dictionary-based approach to mark the spelling of a particular student entry. Instead, it compares the student entry to the Anticipated Correct/Continue Response and reports in its own particularly efficient way.

Note: The spellmarker is case sensitive unless you choose to deselect the Check Case option from the Script Options menu, in which case, WinCALIS will not mark the capitalization errors.

An author may find it very convenient to include the <SPELLMARK> command in the feedback to the student. Consider this example:

?What do you call the thick-skinned nut that grows
?in exotic places
?and has a milk-like liquid in it?
+coconut; Correct
-c*; Check your spelling: <SPCK>

In the example shown above any answer beginning with the letter “c” will be compared by WinCALIS to the first Correct/Continue Anticipated Response and redisplay it with typographic markers to indicate particular spelling errors:

- Extraneous characters will be displayed as ~~strikeout~~ text.
- Missing characters will be displayed as an asterisk (*).
- Wrong characters will be marked with an underline.
- Transposed characters will be displayed in **reverse video** text.
- All four types of mistakes will appear in **bold** type.

For example, the following student entries will be marked as:

Correct/Continue Response:	Student Entry:	Spellmark Feedback:
coconut	coconut	coeconut
coconut	cocnut	coc*nut
coconut	cucunut	<u>c</u> <u>u</u> <u>c</u> <u>u</u> <u>n</u> <u>u</u> <u>t</u>
coconut	cocount	coco <u>un</u> t

If the student entry differs more than 50% from the Correct/Continue Anticipated Response, WinCALIS will display the first character in each word of the Correct/Continue Anticipated Response, followed by a number of underscore characters () equal to the number of remaining characters in the word. For example:

Correct/Continue Response:	Student Entry:	Spellmark Feedback:
coconut	banana	c _ _ _ _ _

In cases in which you have defined an Correct/Continue Anticipated Response using wild cards, comparing the student entry against the Correct/Continue Anticipated Response may make no sense. Neither does it make sense if a student gives an answer which is completely different from the Correct/Continue Anticipated Response. In such cases WinCALIS will display the first letter of the Correct/Continue Anticipated Response. If you would like to include spell-mark information to the student that is appropriate to the answer he typed, you could do so in the following way by using a slightly modified version of the <SPELLMARK> command:

<SPELLMARK: word>

Abbreviation: <SPCK: Word>

?Which modal verb is used in English to give advice?

+{should|ought to}; Correct

-shall; Incorrect

-s*; Check your spelling: <SPCK:should>

-o* to; Check your spelling: <SPCK: ought to>

In the previous example, if a student types an Incorrect/Remain Here response beginning in 's', he or she will get the message "Check your spelling," followed by the answer that was spell-marked against the word "should." The same procedure is followed when the student types an answer beginning in 'o'. In this case the (Incorrect/Remain Here) student answer is marked against 'ought to'. Specifying the word to be spell-marked against is necessary because the Correct/Continue Anticipated Response uses wild cards.

SPELLMARK with Multiple Correct/Continue Anticipated Responses

For those tasks that have more than one Correct/Continue Anticipated Response, WinCALIS provides an automatic feature of Spellmark that automates feedback for "close-but-no-banana" student responses. When a student inputs a response, WinCALIS compares it to all of the Correct/Continue anticipated responses (except those that have wild card characters embedded within), and gives a Spellmarked response based on the one that it is *closest* to. If it differs by more than 50% to all of them, then WinCALIS returns with the first character of each word in the *first* Correct/Continue anticipated response, followed by a number of underscores, representing the number of remaining characters in each word. For example:

Interactive (Question) Window:

Name ONE of the two tree types that were mentioned:

1. Correct/Continue Anticipated Response (Short Answer Task Type):
evergreen

2. Correct/Continue Anticipated Response
deciduous

This task will be handled as follows:

A response of “evergreen” or “deciduous” is considered satisfactory and will be returned with feedback. Any response that is *close* to either of the two target words will be marked accordingly.

Student entry:	Marked as:
evrgreen	ev*rgreen
dissiduous	de ssiduous
oak	e _ _ _ _ _

Executing Another Windows Application

<EXEC: [pathname\]filename.exe[pathname\][file.ext]>

WinCALIS allows you to start other Windows applications from within WinCALIS. This may be convenient for reasons of follow-up or reference. For example, when reading a text, it may be useful to have a card file available for students to make short notes or to consult factual information. After having studied new vocabulary in WinCALIS lessons, a good follow-up task might be to write a paragraph in which the newly-learned words are used. In order to do so, students would need a word processor. The advantages of starting the application from within WinCALIS are obvious: the student could still consult vocabulary in WinCALIS (e.g. Tutorial Windows), while working with the word processor. Since Windows allows for multitasking (having several applications active at the same moment) and sizing of application windows, authors can increase the flexibility and pedagogical value through use of the software. Consider this example:

?For more information about the highlighted
?topics, click on them. If you want to start a
?card file, where you jot down information as you
?read, you may click on the blue button.
?<M:a1><RV><BLUE> __ <BLUE><RV><M:a2>
\$<MOUSE:a1 a2>;<EXEC: cardfile.exe>

In the above example, the Windows Cardfile application is started and kept available for students. Switching between applications is easy in Windows: pressing Alt+Tab will take them back and forth between the two applications.

If you wanted to start the Windows Write word processor with a particular document (DOCUMENT.WRI) opened, you would use a command like:
<EXEC: WRITE FILENAME.WRI>

Author-Supplied Windows Help

The courseware author can use the Windows Help system to supply Windows On-Line Help *.HLP files, available to the student through the Help menu Author Help menu option.

*New Commands***<HELPPFILE: myfile.hlp>**

Sets an author-supplied Windows Help file for Author Help (in same directory as script).

Sample script:

```
#0<NOSC><NOPR>
?<ADD: 1--1>
#end
```

```
#1
?<BLA><CUR:1><CE>If you want to consult the on-line German-English dictionary, drag down
?the Help Menu and click on Author Help.
?<HELPPFILE:gergloss.hlp>
?...

```

<HELPTOPIC: topicid#>

Set topic id number in author-supplied Windows Help file.

<POPHELP>

Pop up author-supplied Windows Help file for Author Help (Help file must be set first with <HELPPFILE: myfile.hlp>).

Sample Script:

```
#0<NOSC><NOPR>
?<ADD: 1--1>
#end

#1
?<BLA><CUR:1>Click on the suitable button for the help you desire.
?Then drag down the Help Menu and click on Author Help.
?<HELPPFILE:authhelp.hlp>
?
?  <BUTTON subverb CREATE "Subject-Verb Agreement",200,25>
?
?  <BUTTON plural CREATE "Noun Plural Formation",200,25>
?
?  <BUTTON inflect CREATE "Verb Inflection",200,25>
?
?  <BUTTON exit CREATE "Exit",200,25>
$<BUTTON subverb>;<HELPTOPIC:2>
$<BUTTON plural>;<HELPTOPIC 3>
$<BUTTON inflect>;<HELPTOPIC 4>
+<BUTTON exit>
```

<POPHELP myfile.hlp [,topicid#]>

Pop up Help file and display topic id#.

Sample script:

```
#0<NOSC><NOPR>
?<ADD: 1--1>
#end

#1<NOSC><NOPR>
?<BLA>
?<CE>Press "s" for Help on subject-verb agreement.
?<CE>Press "n" for Help on noun-plural formation.
?<CE>Press "v" for Help on verb inflection.
?
?<CE>Press ENTER to go on.
+<ENTER>;
=<KEY 's'>;<POPHELP:engsyntax.hlp, 2>
=<KEY 'n'>;<POPHELP:engsyntax.hlp, 3>
=<KEY 'v'>;<POPHELP:engsyntax.hlp, 4>
```

Sequential Flow of the Script through Task List Manipulation

Within WinCALIS Author, the concept of the “task list” remains completely invisible to you. But when you begin to work with WinCALIS scripts and CALIScript, the concept of the task list becomes equated with task flow. The task list tells WinCALIS the order in which it is to work with questions. Normally, we think of this order as beginning at the zero task (#0) and continuing to the end task (#end). But there will be times when you will want to change the order depending on the performance of the student. You might want to give a student extra tasks for review if he or she misses a particular question, or you might want to skip over certain questions if proficiency is demonstrated. By manipulating the “task list,” you can tell WinCALIS to add questions to a script or skip some, based on student performance. Before you learn the practical aspects of task list manipulation, the task list concept will be discussed. To this point, you have learned the basics of the WinCALIS task list:

- All tasks are set off by task labels: the character # followed by the name of the task.
- Any task which requires a student response must contain at least the following two statements: the task label statement, which starts with the pound symbol (#), and a Correct/Continue answer statement, which starts with the plus symbol (+).
- The special task #0 is always the first task executed by WinCALIS. The special task #end is always the last task executed by WinCALIS.

The author may not want to limit the student to completing the tasks that lie between the #0 and #end tasks. You may wish to delete a task which is no longer necessary, to insert tasks if it becomes clear that a student needs help in a particular area or you may wish to add tasks that are to be completed once the first group of tasks is finished. To accomplish all of this, WinCALIS provides the commands <DELETE>, <INSERT> and <ADD>. This will bring a slight change to the way a WinCALIS script has looked to this point.

```
#0
#1
.
.
#2
.
.
#end
```

#extra

Note: The task labeled “extra” occurs after the special task “end.” This means that the task labeled “extra” will never be presented to the student, unless it is brought into one of the task lists using the <INSERT> or <ADD> commands.

When talking about the “task list,” we are actually talking about two lists, the “primary task list” and the “secondary task list.” The primary task list is broadly defined as the set of tasks that the student is working on at the moment. The secondary task list contains the tasks that can be added for completion after the primary task list is finished.

Deleting Tasks from the Task List

<DELETE: task1, task2, ..>

Abbreviation: <DEL: task1, task2, ..>

To delete tasks from any task list (the primary task list or the secondary task list), use the <DELETE> command. It is written as follows:

<DEL: one, two, ..>

where “one” and “two” are examples of names of tasks that might be in the task list. These names must not be written with the task label symbol (#) included. Any number of tasks may be deleted from the script. The only ones which may not be deleted are “end,” and the task which contains the <DELETE> command.

You can specify a range of tasks by giving the label names of two tasks, separated by two hyphens. The command will then affect only those tasks located sequentially in the script starting with the first task in the range and ending with the last task in the range. Note that the label of the first task in the range is separated from the label of the last task in the range with *two* hyphens. For example, the command

<DEL task1--task10>

will delete from the primary task list all the tasks between task1 and task10, inclusive. Do not put spaces before, between, or after the hyphens; the following are all incorrect:

first --last
first-- last
first- -last

Double-check the spelling of all labels. If WinCALIS cannot find a task with the given label, it will do nothing, or it will assume that the range of tasks ends with the last available task.

The <DELETE> command will most commonly occur in the feedback portion of student response statement--after the separator (;) -- in anticipated Correct/Continue, Incorrect/Remain Here, and neutral statements; this links a change in the task list to a specific student response.

If a task occurs more than once in the primary and secondary task lists, <DELETE> will remove all instances of it. If a task is deleted with the <DELETE> command before the student completes it, WinCALIS will behave as if the task was never attempted (even if it was), and that task will neither count positively nor negatively in the student's score. If, at a later point, the task is reinserted, WinCALIS will behave as if the task has been inserted for the first time.

#third-person-pronoun

?Supply the correct form of the pronoun indicated.
?She gave _____ the book. (he)
+him;Yes. "She gave him the book" is correct.
&<DEL: third-person-practice>

#first-person-pronoun
?Supply the correct form of the pronoun indicated.
?Give _____ the book! (I)
+me;Yes. "Give me the book!" is correct.
&<DEL: practice1, practice2, practice3>

In the first task, when students get the answer correct, the further 'third-person' drill practice, which appears to be unnecessary, is deleted. In the second task, three practice tasks are deleted when a student answers correctly.

Inserting Tasks into the Primary Task List

<INSERT: task1, task2, ..>

Abbreviation: <INS: task1,task2, ..>

To insert tasks into the primary task list, use the <INSERT> command. It is written as follows:

<INS: one, two, ..>

where "one" and "two" are examples of the names of tasks which appear somewhere in the script. These names must not be written with the task label symbol (#) included. Any number of tasks may be inserted into the primary task list.

You can specify a range of tasks by giving the label names of two tasks, separated by two hyphens. The command will then affect those tasks located sequentially in the script starting with the first task in the range and ending with the last task in the range. Note that the label of the first task in the range is separated from the label of the last task in the range with *two* hyphens. For example, the command

<INS task1--task10>

will insert into the primary task list all the tasks between task1 and task10, inclusive. Do not put spaces before, between, or after the hyphens; the following are all incorrect:

first --last
first-- last
first- -last

Double-check the spelling of all labels. If WinCALIS cannot find a task with the given label, it will do nothing, or it will assume that the range of tasks ends with the last available task.

The <INSERT> command will most commonly occur in the feedback portion of student response statement, after the separator (;) in Correct/Continue, Incorrect/Remain Here, and neutral answer statements. Thus <INSERT> links a change in the task list to a specific student response.

<INSERT> will behave differently depending upon whether it occurs in a Correct/Continue or Incorrect/Remain Here answer statement. When <INSERT> occurs in a Correct/Continue answer statement, the current task is marked correct, the score is recorded, and the student proceeds to the task at the head of the list (which has just been inserted into that position). When <INSERT> occurs in an

anticipated Incorrect/Remain Here statement, the current task is immediately suspended. WinCALIS behaves as if the current task has been passed over for the moment. The student proceeds to the task at the head of the list (which has just been inserted). After all inserted tasks have been completed the student will be returned to the original task. For example,

```
#third-person-pronoun
?Supply the correct form of the pronoun indicated.
?She gave _____ the book. (he)
+him;Yes. "She gave him the book" is correct.
&<INS: next-pronoun>
-he|his;Another pronoun is needed. Complete the
&following practice questions. You may return to
&this question later.
&<INS: practice1, practice2, practice3>
```

The Correct/Continue answer statement accepts the student's answer and inserts a new task at the front of the task list. The student will move on to the task #next-pronoun, and, from there, to whatever task originally followed #third-person-pronoun. The anticipated Incorrect/Remain Here statement inserts three practice drills. The student will complete these three, and then will return to #third-person-pronoun to complete it.

Adding Tasks to the Secondary Task List

<ADD: task1, task2, ..>

no abbreviation

To add tasks to the secondary task list, use the <ADD> command. It is written as follows:

<ADD: task1, task2 ..>

where "one" and "two" are examples of the names of tasks which appear somewhere in the script. These names must not be written with the task label symbol (#) included. Any number of tasks may be added to the secondary task list.

You can specify a range of tasks by giving the label names of two tasks, separated by two hyphens. The command will then affect those tasks located sequentially in the script starting with the first task in the range and ending with the last task in the range. Note that the label of the first task in the range is separated from the label of the last task in the range with *two* hyphens. For example, the command

<ADD task1--task10>

will add to the secondary task list all the tasks between task1 and task10, inclusive. Do not put spaces before, between, or after the hyphens; the following are all incorrect:

```
first --last
first-- last
first- -last
```

Double-check the spelling of all labels. If WinCALIS cannot find a task with the given label, it will do nothing, or it will assume that the range of tasks ends with the last available task.

The <ADD> command will most commonly occur in the feedback portion of student response statement, after the separator (;) in Correct/Continue, Incorrect/Remain Here, and neutral answer statements. Thus <ADD> links a change in the task list to a specific student response.

The secondary task list is created using <ADD> commands. <ADD> appends tasks to the end of the secondary task list. Added tasks are not available to students immediately, but are delayed until all tasks in the primary task list are completed. Once a student finishes the tasks in the primary task list, WinCALIS moves to the secondary task list, which becomes the new primary task list. This is useful, for example, when an author wants to postpone detailed explanation of a student's mistake until later, or when performing a task which must occur at the end of a lesson. Here is an example of the use of the <ADD> command in a task:

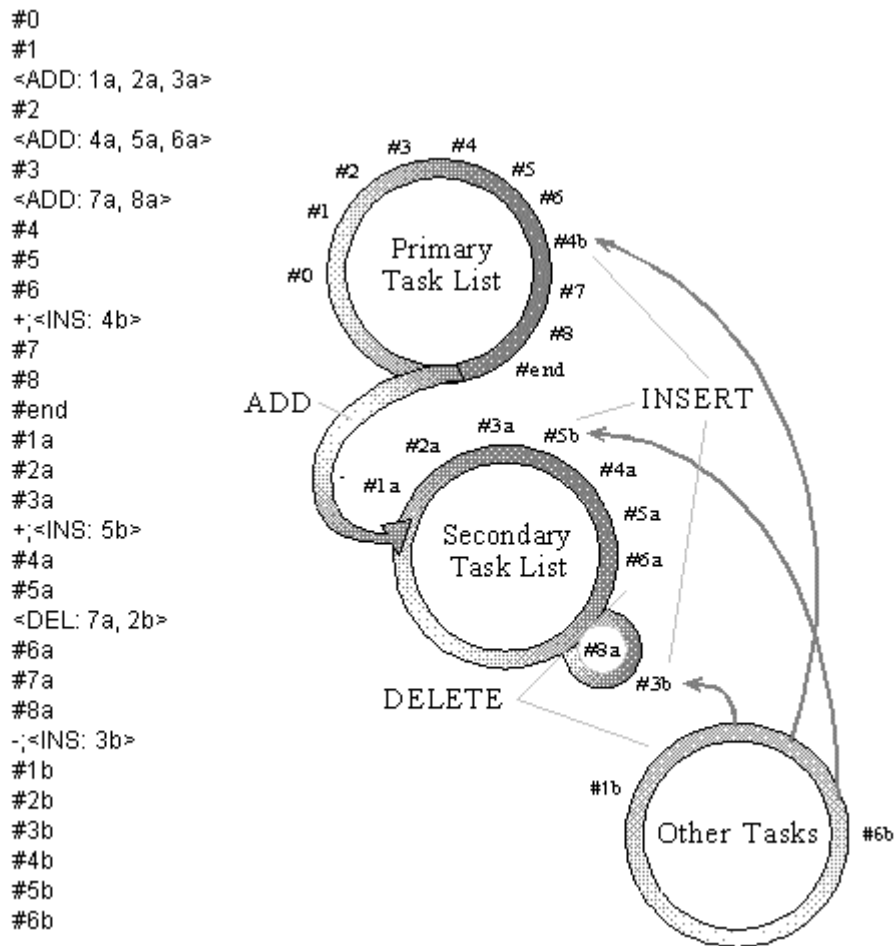
```
#third-person-pronoun <NOSCORE>
?Supply the correct form of the pronoun indicated.
?She gave _____ the book. (he)
+him;Yes. She gave him the book.
-{he|his};<ADD: third-person-practice>
-&;<ADD: drill1, drill2, third-person-practice>
```

The sample script shown above might be useful in a lesson on pronouns. The initial task list would contain several questions that test a student's competence with several forms of pronouns. A more comprehensive task list would then be assembled, which would start when the student finished the initial questions. Here there is simple reinforcement for the correct answer. An Incorrect/Remain Here answer causes the task “#third-person-practice” to be added to the secondary task list. An unanticipated response adds the practice plus two drill tasks. Notice that each of the three answers occurs in a correct answer line. This task is not scored. (<NOSCORE> is discussed in the section Task Options and Defaults).

You may wonder why the tasks to be added cannot simply be put some place after the original anticipated Correct/Continue responses in the original (primary) task list, and deleted if the student gets the corresponding question correct. This isn't possible because of the existence of the Pass keys; there is no guarantee that the student will complete the questions in the order presented by the author.

An Example: Primary and Secondary Task Lists

The following diagram presents the outline of a moderately complex WinCALIS script. If you are comfortable with the concept of “branching,” the flow of the diagram should become clear. The diagram shows the order in which a WinCALIS script completes tasks. The sample script shows the order in which the tasks appear in the author's script. The diagram shows the order in which the tasks are actually completed by the student.



The diagram above shows how the tasks presented in the sample script, are actually presented to and completed by the student. The primary and secondary task list are presented in a circular form. This is to facilitate the use of the Pass forward and Passback options. WinCALIS does not quit at the end of a task list, but presents the first task (and all of the following) again, thus making the circle complete. Only after a student has successfully completed all of the tasks in a task list (or has given up by pressing the F6 key), will there be no more tasks left in the circular task list.

The point at which inserted tasks appear in a script depends on if they were inserted into Correct/Continue or Incorrect/Remain Here student responses. In case a task is inserted following a Correct/Continue student response, the task will be presented following the task it was initiated by. In case a task is inserted following an Incorrect/Remain Here student response, the current task will be passed and the inserted task will be presented first. After completing the inserted task, WinCALIS will return to the task it was initiated from, thus having made a loop.

Implementing a WinCALIS Task Database

With WinCALIS it is possible to create a set of questions, similar in degree of difficulty, and randomly present a certain number of them to the student. In this way drills can be varied or students can be given different exams drawn from the same task database, or "taskbase," a large set of questions (in WinCALIS parlance, "tasks") that can be accessed randomly and reused for different students.

See Also: Sequential Flow of the Script through Task List Manipulation.

Inserting and Adding Tasks at <RANDOM>

<RANDOM [n] INSERT|ADD [GLUE] [r1, rn]>

Abbreviation: <RAND [n] INS|ADD [GLUE] [r1, rn]>

Options:

number of tasks ([n]): the number of tasks to be inserted or added to the task list type

(INSERT|ADD): either insert or add the tasks to the task list source type

([GLUE]): tasks that have been glued to the task list from another file range of tasks

([r1, rn]): delineates the range from which tasks are to added or inserted.

The <RANDOM> command will randomly insert or add a task. It will choose a task at random from those which are not included in the primary or secondary task list and will insert that task into the primary (in the case of an 'insert' procedure) or secondary (in the case of an 'add' procedure) task list.

The tasks which are not part of any task list are those which were originally before task #0 or after task #end, or which were subsequently inserted or added (by means of the <INSERT> or <ADD> commands) to a task list. Once a task has been randomly inserted or added it will not (and can not) be randomly selected again in that lesson. Look at the following example:

```
#0
.
#1
.
#2
.
#end

#A
.
#B
.
```

In the above example, tasks #0, #1, #2 and #end may not be added or inserted randomly as they are part of the original primary task list. Tasks #A and #B may be. The <RANDOM> command must specify whether the randomly selected task is to be inserted or added:

- The command <RAND INS> will insert one randomly selected task into the primary task list.
- The command <RAND ADD> will add one randomly selected task to the secondary task list.

The <RANDOM> command can contain a number of options:

You may specify a certain number of tasks. By default, the <RANDOM ADD|INSERT> command will add or insert randomly one task to the appropriate task list. You can add or insert more than one task by specifying the number of tasks to be inserted or added. For example, the command

<RAND 10 INSERT>

will insert ten randomly selected tasks into the primary task list. If you give a number which is greater than the amount of available tasks, WinCALIS will add or insert all available tasks.

You may specify 'glued' tasks. If the author includes the word "GLUE" in the <RANDOM> command, WinCALIS will randomly add or insert only from those tasks which have been glued to the current script file with the <GLUE> command, and will ignore all others. The <GLUE> command will be discussed below (See the section "<GLUE>ing Tasks") For example, the command

<RAND ADD GLUE>

will add one task to the primary task list from among those which have been glued to the current task list with the <GLUE> command. This is helpful, as 'glued' tasks can be specifically designed to work with the <RANDOM> command. The <GLUE> command is needed to make use of a large randomly-accessed task database.

You can specify a range of tasks. The <GLUE> option is one way to narrow the range of tasks from which WinCALIS makes its random selection. A more specific way of delineating a range is by giving the label names of two tasks. WinCALIS will then choose only from among those tasks that are located after the first task in the range but before the last task in the range. (The first and last tasks, the ones whose label are given, are also included in this range.) The label of the first task in the range is separated from the label of the last task in the range with *two* hyphens. For example, the command

<RAND INS voc1--voc10>

will insert one task into the primary task list from among those located between the tasks labeled (voc1 and voc10 inclusive). Do not put spaces before, between, or after the hyphens; the following are all Incorrect/Remain Here:

first --last
first-- last
first- -last

Because only tasks that are not part of any task list may be randomly added or inserted, be sure not to use any tasks that are originally located between tasks 0 and end. Also double check the spelling of all labels. If WinCALIS cannot find a task with the given label, it will do nothing, or it will assume that the ranges of tasks ends with the last available task.

Options can be mixed in one <RANDOM> command. For example:

<RAND 20 INS GLUE noun10--noun20>

will insert twenty tasks into the primary task list from among those "glued on" with the <GLUE> command.

<RAND 10 ADD noun10--noun30>

will add ten tasks to the secondary task list from among those located between the tasks labeled "noun10" and "noun30."

<RAND 30 ADD noun10--noun30>

will add twenty-one tasks to the secondary task list, that is, all tasks between those labeled noun10 and noun30, including tasks noun10 and noun30. (WinCALIS cannot add thirty tasks, as there were only twenty-one tasks in this hypothetical range).

How to GLUE Tasks

<GLUE: filename.ext>

no abbreviation

The <GLUE> command “glues on” additional WinCALIS tasks to the original WinCALIS script. In a task database, the <GLUE> command allows the <RANDOM> function to choose a few WinCALIS tasks from among hundreds). For example, the command <GLUE: randfile> will attach the WinCALIS task database, or “taskbase”, “randfile” to the end of the WinCALIS script.

The “glued” file must be a valid WinCALIS taskbase file, i.e. it must be a binary Unicode script file, having the same format as a *.WCL file that WinCALIS creates when it runs a *.CAL file for the first time. (See the section Relationship between WinCALIS and CALIScript.) However, it does not need to have the extension .WCL. If you generate script files that you intend to use exclusively as taskbase files, it is even better to change the .WCL extension into another, e.g. .GLU. The advantage is that if the extension .WCL is changed to .GLU, the taskbase script will not show up in the Lesson Window in WinCALIS (when the option “No Conversion *.WCL is selected from the Author Workstation Options menu), since it does not have the required .WCL extension. As a consequence, it is recommended that if you want to produce a taskbase file, you follow this procedure:

- Use WinCALIS Author to produce a taskbase file.
- Run the lesson once in WinCALIS.
- Use the Windows File Manager to rename the WinCALIS-generated “filename.wcl” file into a “filename.glu” file
- Insert the <GLUE: filename.glu> command in the script file into which you want the “filename.glu” taskbase file to be glued.

Once attached with the <GLUE> command, the tasks from the taskbase file can be brought to the student with the <INSERT>, <ADD>, and <RANDOM> commands. As such, they are like any task which is placed outside of the tasks #0 and #end in the original WinCALIS script.

```
#0
.
#1
?<GLUE: randfile.glu>
+correct response;<RANDOM 2 INSERT GLUE>
#2
.
#end
```

In the above example, the WinCALIS taskbase file “randfile.glu” containing the tasks #RF1, #RF2, and #RF3, is glued to the original script. The effect of this is to produce what may be viewed as a revised script such as the one shown:

```
#0

#1
?<GLUE: randfile.glu>
+correct response;<RANDOM 2 INSERT GLUE>
#2
.
#end
```

```
#RF1
.  
#RF2
.  
#RF3
```

In the above example, which contains the same script as the former example, the glued tasks from `randfile.glu` have been specified. When the student responds correctly in task #2, two randomly selected tasks from the glued tasks (#RF1, #RF2, #RF3) will be inserted in the primary task list. The use of the `<GLUE>` command and prepared taskbases in WinCALIS has two advantages:

- Different scripts may make use of the same task database. This keeps an author from having to duplicate the same tasks in different WinCALIS scripts (and from having to edit many files if a change is made).
- Many more potential tasks can be loaded into WinCALIS with the `<GLUE>` command than can be loaded from the start, using the same amount of computer memory (RAM). This is because the `<GLUE>` command adds only information about the tasks in the taskbase file to WinCALIS. This conserves memory. It adds all of the task structure after the task has been called with the `<INSERT>`, `<ADD>`, or `<RANDOM>` command.

There is one disadvantage to the use of WinCALIS task databases and the `<GLUE>` command. Because only information about the tasks in the taskbase file is glued on at first, WinCALIS must again access the taskbase file on disk when a glued task is called with the `<INSERT>`, `<ADD>`, or `<RANDOM>` command. This may be a little slower than getting the task directly from the computer's memory (RAM).

The labels given the tasks in the taskbase are important. As these tasks will be brought into a running WinCALIS script, one should avoid using a label name that may conflict with a task in that running WinCALIS script. Obvious labeling schemes (for example, using consecutive numbers , #1, #2, #3, etc.) must be avoided. One good idea is to put a prefix or suffix on each task label that would identify it with a particular taskbase. For example, a taskbase that contains only tasks that have to do with relative clauses might have tasks labeled #RC1, #RC2, etc.

The task options associated with the glued tasks (e.g., `<NOBLANK>`, `<NOSCORE>`) will be the ones they had when the WinCALIS taskbase file was run for the first time (as a *.CAL file). Any global task options in the running WinCALIS script will not affect them. For example, if the running WinCALIS script looks like the following:

```
@<BLANK>
#0
.  
#1
?<GLUE: randfile.glu>
#2
.  
#end
```

and the taskbase script, `randfile.glu`, to be glued to the running script looks like this:

```
@<NOSCORE><NOBLANK>
#RF1
.  
#RF2
.
```

#RF3

the script, after the “randfile.glu” has been glued to it, will run as follows:

```
@<BLANK>
#0
#1
?<GLUE: randfile.glu>
#2
.
#end

@<NOSCORE><NOBLANK>
#RF1
.
#RF2
.
#RF3
```

The glued tasks from “randfile.glu” have their original task options, <NOSCORE> and the default task option <NOBLANK>.

Any number of taskbases may be glued to a WinCALIS script. However, use only one taskbase filename per <GLUE> command. For example, to glue on the taskbases “newfile.glu” and “newfile2.glu” issue two commands, <GLUE: newfile.glu> and <GLUE: newfile2.glu> separately.

How to UNGLUE Tasks

<UNGLUE: filename.glu>

Abbreviation: <UNG: filename.glu>

As its name implies, <UNGLUE> removes task information that had been inserted with the <GLUE> command. It is a housekeeping command that helps to avoid the problem of having two tasks with the same label conflict with each other.

After the <UNGLUE> command is issued, tasks previously added with the <GLUE> command can no longer be accessed. It offers a clean slate for subsequent <GLUE> commands.

<UNGLUE> removes only the task information that had been inserted with <GLUE>; it does not remove any actual tasks that were subsequently accessed with the <INSERT>, <ADD>, or <DELETE> commands. For example, in the following sample script, the author brings in a taskbase file called “newfile1.glu” with the <GLUE> command, and randomly selects ten tasks from it for insertion into the task list. As the author no longer plans to use any tasks from “newfile1.glu,” the <UNGLUE> command is issued. The taskbase file “newfile2.glu” is brought in, and the ten tasks randomly inserted with the subsequent <RANDOM> command, are all from the taskbase file “newfile2.glu.”

```
#0
#1
?<GLUE: newfile1.glu><RANDOM 10 INSERT GLUE>
.
#2
?<UNGLUE:newfile1.glu><GLUE:newfile2.glu> <RANDOM 10 INSERT GLUE>
```

Dynamic Commands: Special Considerations

The dynamic commands include <DELAY> and <BEEP>. The <AUDIO> command is dynamic, as well as the <VIDEO> command. They are discussed in the section Media Commands. If any of these commands are directed to a window which is not on the screen, they and any subsequent text or commands sent to that window are put on hold until that window is displayed to the student. Therefore, an author must be sure not to put any commands in these windows which WinCALIS is meant to execute immediately.

The Interactive (Question) Window and Presentation (Text) Window, if one exists, are always treated as if they are displayed. No hold is ever created for them. Basically, we are concerned with the windows that are popped up or can be popped up by the student: the Directions, Feedback Diagnostic and F1-F10 Tutorial Windows.

Commands that redirect WinCALIS (<DIR>, <Q>, <T>, <D>, <F1>...<F10>) to other windows are never put in hold, but instead are executed immediately. Use them freely to keep your lessons in order.

On the other hand, WinCALIS commands that act on another window without actually switching to it (e.g., <BLANK F8>, <CUR: DIR marklabel>) are put on hold and will not be executed until the hold window is accessed by the student.

For example, if the command <CLEAR: F8> appears in a series of commands being sent to the F7 window, and F7 had previously been issued a dynamic command, the command <CLEAR: F8> would not be executed until the F7 window was opened (either by the author, or by the student pressing the Shift+F7 key combination). If the author wishes the F8 window to be cleared before this, the F8 window should be accessed directly: <F8><CLEAR>.

The <BLANK> command is somewhat of an exception. If a <BLANK> command is sent to a window on hold, it is acted upon immediately: the window and the hold are blanked (even if the student has yet to see the contents). However, if it is meant to act on another window the command will be kept on hold until the window it is written to is opened. For example, if the command <BLANK F8> is directed to the F7 window, and the F7 window had previously been issued a dynamic command, the <BLANK F8> command would remain on hold until the F7 window was opened.

Special Responses

In WinCALIS, certain keys have a special significance. Pressing the F9 key will pop up the latest Popup (Directions) Window; pressing the Shift key in combination with one of the first ten function keys, allows the student to consult a Tutorial Window. Four keys that the student will use often are the ENTER key (sometimes called RETURN or carriage return), the Giveup key (F6) and the Pass keys (the gray + and - keys on the numeric keypad of the keyboard or Ctrl+PageDown and Ctrl+PageUp). Moreover, in WinCALIS authors have the option of specifying tasks in which students respond by clicking the mouse. And if necessary key combinations can be given a special status. When authors need to present special feedback in conjunction with these particular keys (the mouse or a defined key combination) special responses can be included in anticipated Correct/Continue, Incorrect/Remain Here, and neutral answer lines. This section explains how to use the special responses.

Note: All CALIS commands, options, and special responses are composed of one or more key words in capital letters, surrounded by opening and closing angle brackets (<>), and separated by one or more delimiters.

Pressing a Button

<BUTTON name [CLICKED|CHECKED|UNCHECKED]>

no abbreviation

Defines action by WinCALIS if student “presses” (clicks, checks, doesn't check) hot button created with <BUTTON> command. (See Creating Buttons and Hotspots.)

Simple example:

```
#button
?<BLA>This shows how to add a hot button to a window.
?
?A WinCALIS lesson file is often called a _____.
?
? Press here if you don't know => <BUTTON clue CREATE “Clue”>
+script
=<BUTTON clue>;Every play or movie has one of these,
&and so does WinCALIS.
```

<BUTTON name1 DRAGDROP name2>

no abbreviation

Defines action by WinCALIS if student drags and drops hot button name1 created with <BUTTON> command onto hot button name2.

Simple example:

```
#buttons
?<BLA><CUR:2><CE>Drag the names on the left over to their correct reigns on the right.
?<CUR:5,20><BUTTON: Aa CREATE “Versailles” STATIC 150 30>
?<CUR:5,70><BUTTON: Bb, CREATE, “Louis XVI”, STATIC, 150, 30>
?<CUR:8,20><BUTTON: Ba, CREATE, “Marie Antoinette”, STATIC, 150, 30>
?<CUR:8,70><BUTTON: Db, CREATE, “Charles VII”, STATIC, 150, 30>
?<CUR:11,20><BUTTON: Ca, CREATE, “Manet”, STATIC, 150, 30>
?<CUR:11,70><BUTTON: Ab, CREATE, “Louis XIV”, STATIC, 150, 30>
?<CUR:14,20><BUTTON: Da, CREATE, “Joan of Arc”, STATIC, 150, 30>
?<CUR:14,70><BUTTON: Cb, CREATE, “Napoleon III”, STATIC, 150, 30>
?<BUTTON: Aa, FCOLOR, WHITE><BUTTON: Aa, BCOLOR, DKBLUE>
?<BUTTON: Ab, FCOLOR, WHITE><BUTTON: Ab, BCOLOR, DKBLUE>
?<BUTTON: Ba, FCOLOR, WHITE><BUTTON: Ba, BCOLOR, DKBLUE>
?<BUTTON: Bb, FCOLOR, WHITE><BUTTON: Bb, BCOLOR, DKBLUE>
?<BUTTON: Ca, FCOLOR, WHITE><BUTTON: Ca, BCOLOR, DKBLUE>
?<BUTTON: Cb, FCOLOR, WHITE><BUTTON: Cb, BCOLOR, DKBLUE>
?<BUTTON: Da, FCOLOR, WHITE><BUTTON: Da, BCOLOR, DKBLUE>
?<BUTTON: Db, FCOLOR, WHITE><BUTTON: Db, BCOLOR, DKBLUE>

#dragdrop<NOKEY><ALL>
+<BUTTON: Aa DRAGDROP Ab>;<CE>Yes, it was Louis XIV's palace.
-<BUTTON: Aa DRAGDROP Bb>;<CE>Wrong Louis!
-<BUTTON: Aa DRAGDROP Cb>
-<BUTTON: Aa DRAGDROP Db>
-<BUTTON: Ba DRAGDROP Ab>;<CE>Wrong Louis!
+<BUTTON: Ba DRAGDROP Bb>
-<BUTTON: Ba DRAGDROP Cb>
-<BUTTON: Ba DRAGDROP Db>
-<BUTTON: Ca DRAGDROP Ab>
-<BUTTON: Ca DRAGDROP Bb>
+<BUTTON: Ca DRAGDROP Cb>
```

```
-<BUTTON: Ca DRAGDROP Db>
-<BUTTON: Da DRAGDROP Ab>
-<BUTTON: Da DRAGDROP Bb>
-<BUTTON: Da DRAGDROP Cb>
+<BUTTON: Da DRAGDROP Db>;Correct. She was burned in 1431.
```

The Enter Key

<ENTER>

no abbreviation

The ENTER key is in some respects the most important key in WinCALIS; it is the one students press when they are finished typing their answers. ENTER sends the student response to WinCALIS for processing. This basic function never changes. What we will deal with here are those cases in which the student presses the ENTER key without having typed in any response.

When students type the ENTER key without any other response WinCALIS simply passes on to the next question. WinCALIS acts exactly as if the Pass Forward key (the gray + or Ctrl+PageDown) had been typed. Unless the <ENTER> response is specified, pressing the ENTER key without typing any response is the same as pressing the Pass Forward key (gray + or Ctrl+PageDown). This feature is simply a convenience for students.

Note: As a convenience for laptop users, WinCALIS now also supports the additional Pass Forward key Ctrl+PageDown and the additional Pass Back key Ctrl+PageUp.

There are two instances in which an author may want the ENTER key to behave differently. First, when a task merely presents information, students can signal their readiness to continue by typing the ENTER key in lieu of a more elaborate response. In this circumstance the <ENTER> response should be placed on a Correct/Continue answer (+) line, along with any desired feedback. Consider this example:

```
#all-done
? Here is your completed chart. This concludes Part 1.
? Press [Enter] to continue.
+<ENTER>
```

When students press the ENTER key, WinCALIS proceeds to the next task. A second situation where you may specify a different response for <ENTER> is in an anticipated Incorrect/Remain Here answer statement. Used here an author is instructing WinCALIS not to pass when a student presses the ENTER key without offering a further response, but instead to treat ENTER like any other anticipated Incorrect/Remain Here answer. This may be useful, for example, if an author wants to use ENTER to trigger some clue or hint.

```
#enter
? In what year did Columbus discover America?
? (Press [Enter] for a hint)
+1492;Very good
=<ENTER>;Hint: Columbus sailed the ocean blue,
&in fourteen hundred and ____ ____.
```

The Giveup Key

<GIVEUP>

no abbreviation

By default, when the student has made at least one attempt at a question and presses the Giveup key F6, WinCALIS displays the first Correct/Continue answer listed for that task in a pop-up window and moves on to the next question. Sometimes, however, the author may want WinCALIS to do more when the student gives up. For example, you may want to rewrite the screen in some way before going on to the next question. In cases such as this the author may redefine the F6 Giveup key with the <GIVEUP> special response.

```
#giveup
/<BLA>
/<CE>___ am happy.
?<BLA>
?<CE>What pronoun completes the above sentence?
+I;<T><BLA><CUR:2>,<CE>I am happy
&<D>Very good. This is correct.
-<GIVEUP>;<T><BLA><CUR:2>,<CE>I am happy
&<D>Correct Answer: I
#moveon
?<BLA><CE><CUR:1>
?<CE>Now for the next step...
```

In the above example, the author has decided to rewrite the sentence in the Presentation (Text) Window after the student's response. It originally appears like this:

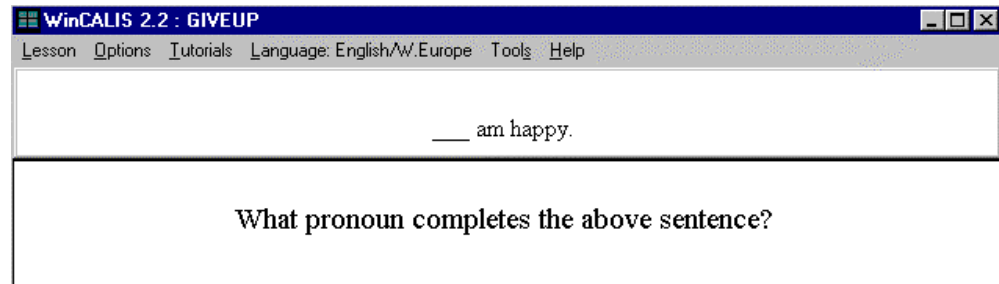


Figure 15: Question as originally presented

After giving up on the question in the above example the student sees the Correct/Continue answer and moves on to the next task. The sentence in the Presentation (Text) Window has been rewritten to show the Correct/Continue answer.

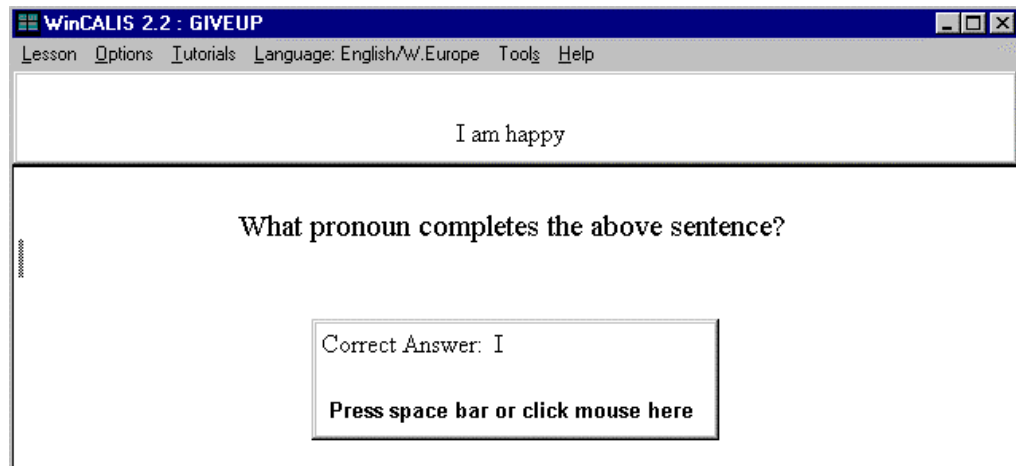


Figure 16: After a Give Up response, sentence is rewritten through feedback

Without using the <GIVEUP> special response, the author would not have been able to overwrite the sentence, “I am happy,” in the Presentation (Text) Window if the student gave up.

The <GIVEUP> special response should be used only in anticipated Incorrect/Remain Here (-) statements.

Making Hotspots in Graphics

<HOTSPOT name [0,0,100,100]>

no abbreviation

In order to identify a hotspot on a graphic, you first need to give the graphic a “handle” or name by which you refer to it in the HOTSPOT special response. The name can be simply the 1-8 letter filename for the *.BMP file, or any other string of letters you choose.

For example, in the sample script shown under the <HOTSPOT> special response, the graphic “durham.bmp” is given the name “durham” with the command

<GRAPHIC: durham.bmp durham>

You can then use the <HOTSPOT> special response to define a “hot” area on the graphic, which, if clicked on with the mouse, will produce a specific effect. By default the hotspot is 100% of the graphic. The optional parameters restrict the hot spot to a smaller area.

The parameter conventions are the same as for the <SIZE> and <VIDEO> location commands, but here the reference points are the top and left side of the entire graphic, rather than the screen. For example, in the special response <HOTSPOT 20,30,40,60>, you can think of the first two numbers, 20 and 30, as defining the upper left corner of the hotspot to be down 20 per cent from the top of the graphic and over 30 percent from the left side of the graphic, and the latter two numbers as defining the lower right-hand corner to be down 40 per cent and over 60 per cent. Alternatively the parameters may be thought of as indicating the per cent down from the top of the graphic and over from the left side of the top, left side, bottom, and right side of the hotspot.

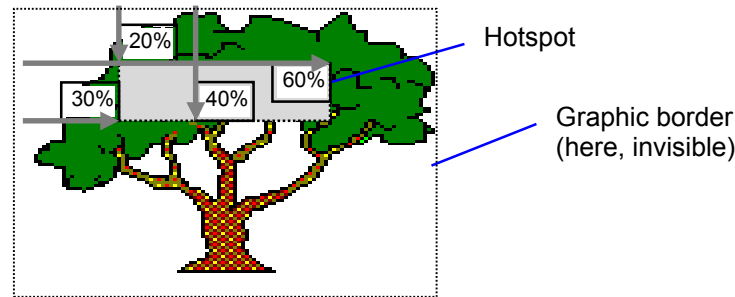


Figure 17: Indicating a hotspot on a graphic <HOTSPOT 20, 30, 40, 60>

As many hotspots as you want can be defined for the same graphic. They may also be overlapping. Since hotspots are always rectangles, a series of small overlapping rectangular hotspots can be used to define a slanting or angular hotspot.

Sample script:

#0

!You will be shown a small section of a map of Durham.

!Click on the streets or places indicated below the map.

#map

?<SET FC DKRED><CUR:2,28><GRAPHIC:durham.bmp durham>

#1

?<CUR:16>Click on Club Boulevard.

+<HOTSPOT durham 7,0,17,100>;Yes, this is Club Blvd.

-<HOTSPOT durham 0,78,100,83>;No, this is Gregson St.

-<HOTSPOT durham 0,85,100,90>;No, this is Duke St.

-<HOTSPOT durham 0,45,70,50>;No, this is Broad St..

-<HOTSPOT durham 20,0,35,25>;No, this is Business 70.

-<HOTSPOT durham 35,10,55,40>;No, this is Business 70.

-<HOTSPOT durham 50,35,70,40>;No, this is Business 70.

-<HOTSPOT durham 36,50,80,68>;No, this is Duke's East Campus.

-<HOTSPOT durham 80,0,86,30>;No, this is Erwin Road.

-<HOTSPOT durham 70,25,80,40>;No, this is Erwin Road.

#3

?<CUR:16>Click on Duke's East Campus.

-<HOTSPOT durham 7,0,17,100>;No, this is Club Blvd.

-<HOTSPOT durham 0,78,100,83>;No, this is Gregson St.

-<HOTSPOT durham 0,85,100,90>;No, this is Duke St.

-<HOTSPOT durham 0,45,70,50>;No, this is Broad St..

-<HOTSPOT durham 20,0,35,25>;No, this is Business 70.

-<HOTSPOT durham 35,10,55,40>;No, this is Business 70.

-<HOTSPOT durham 50,35,70,40>;No, this is Business 70.

+<HOTSPOT durham 36,50,80,68>;Yes, this is Duke's East Campus.

-<HOTSPOT durham 80,0,86,30>;No, this is Erwin Road.

-<HOTSPOT durham 70,25,80,40>;No, this is Erwin Road.

#end

Acting on a Special Response IF a Condition is Satisfied

<IF:response\$=variable\$>

no abbreviation

Defines action by WinCALIS if the student's text response, stored in response\$, matches the value of variable\$,

<IF:mark\$=variable\$>

no abbreviation

or text marked, stored in mark\$, matches the value of variable\$,

<IF:numvariable=value>

no abbreviation

or expression numvariable=value is true.

<IF:numvariable LT value>

no abbreviation

or expression numvariable<value is true.

For now, see: Variables

Promoting the Status of a Particular Key (Combination)

<KEY 'x'>

no abbreviation

Usually the student must press the ENTER key after typing in an answer in order to a response from WinCALIS. However, with the special response command, <KEY: 'x'>, it is possible for the author to designate a keystroke which is interpreted by WinCALIS immediately without being followed by ENTER. Consider the following example:

?Press "B" to begin.

+<KEY 'b'>;<ADD: 1,2,3>

As soon as the student presses the "b" key tasks #1, #2, and #3 are added to the task list using the <ADD> command and the current task is completed. (Please refer to the section Sequential Flow of the Script through Task List Manipulation.) Combination keystrokes may also be defined with this special response command using the Shift and Ctrl keys. All of the following are valid:

<KEY:SHIFT 'b'>

<KEY:CONTROL'5'>

<KEY:CONTROL SHIFT 'x'>

Remember that WinCALIS special responses are evaluated before other anticipated student responses. Be careful not to combine <KEY:'x'> special responses with anticipated text answers that begin with the same letter. For example:

```
+sure;bad
+<KEY:'s'>;worse
```

As soon as the student presses the “s” key, WinCALIS matches the input with the <KEY:'s'> special response. It is impossible for the student to type “sure” or any word beginning with an “s.”

There are also the following extensions of the <KEY> command:

<KEY ANY>

This will take effect if any key at all is pressed.

<KEY controlkey>

This variation enables you to specify virtually any of the “control” keys on the keyboard. Substitute one of the keywords in the list below for the word 'controlkey' in the command.

SHIFT, CONTROL, ALT, BACK, TAB, CAPITAL, ESCAPE, SPACE, PRIOR, NEXT, END, HOME, LEFT, UP, RIGHT, DOWN, DELETE, NUMPAD0, NUMPAD1, NUMPAD2, NUMPAD3, NUMPAD4, NUMPAD5, NUMPAD6, NUMPAD7, NUMPAD8, NUMPAD9, MULTIPLY, ADD, SEPARATOR, SUBTRACT, DECIMAL, DIVIDE, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, NUMLOCK

The following example script is a menu template, in which the student presses the up and down arrows to move among the menu options.

```
#0
!Menu template
!Search for “Menu Item 1” etc. and replace with your specific menu texts.
!Then search for “body1” etc. and replace with tasks to be executed (e.g., 1--10)
```

```
#menu1<NOSC><NOPR><REP><NOKEY>
?<CUR:6>
?                <RV>Menu Item 1<RV>
?                Menu Item 2
?                Menu Item 3
?                Menu Item 4
?                Exit
=<KEY: DOWN>;<PASS:1>
=<KEY:UP>;<PASSBACK:1>
=<ENTER>;<INSERT:body1>
```

```
#menu2<NOSC><NOPR><REP><NOKEY>
?<CUR:6>
?                Menu Item 1
?                <RV>Menu Item 2<RV>
?                Menu Item 3
?                Menu Item 4
?                Exit
=<KEY: DOWN>;<PASS:1>
=<KEY:UP>;<PASSBACK:1>
=<ENTER>;<INSERT:body2>
```

```
#menu3<NOSC><NOPR><REP><NOKEY>
?<CUR:6>
?           Menu Item 1
?           Menu Item 2
?           <RV>Menu Item 3<RV>
?           Menu Item 4
?           Exit
=<KEY: DOWN>;<PASS:1>
=<KEY:UP>;<PASSBACK:1>
=<ENTER>;<INSERT:body3>
```

```
#menu4<NOSC><NOPR><REP><NOKEY>
?<CUR:6>
?           Menu Item 1
?           Menu Item 2
?           Menu Item 3
?           <RV>Menu Item 4<RV>
?           Exit
=<KEY: DOWN>;<PASS:1>
=<KEY:UP>;<PASSBACK:1>
=<ENTER>;<INSERT:body4>
```

```
#menu5<NOSC><NOPR><REP><NOKEY>
?<CUR:6>
?           Menu Item 1
?           Menu Item 2
?           Menu Item 3
?           Menu Item 4
?           <RV>Exit<RV>
=<KEY: DOWN>;<PASS:1>
=<KEY:UP>;<PASSBACK:1>
+<ENTER>;<DEL menu1--menu4>
#end
```

```
#body1
?<BLA>chunk1
+ok
```

```
#body2
?<BLA>chunk2
+ok
```

```
#body3
?<BLA>chunk3
+ok
```

```
#body4
?<BLA>chunk4
+ok
```

For <PASS:n> and <PASSBACK:n> commands, see New WinCALIS Features and Extensions to CALIScript (Outline).

Clicking the Mouse

Using WinCALIS, you can design a task that the student completes by using the mouse to point and click on a region of the screen which you have previously defined, known as a hotspot. This is done using the <MARK:x> and <MARK:y> commands and the special response:

<MOUSE x y>

no abbreviation

Parameters (x y): Parameters x and y should refer to the upper left-hand and lower right-hand corners of the region in which a student may point and click to give an answer. These corners are defined by means of the <MARK:x> command.

WinCALIS will allow an author to define text as a hotspot, i.e., a region that can be clicked with the mouse as a student response. The upper left and lower right corners are specified by means of a <MARK: x> and <MARK: y> special response. Consider the following example:

```
#mouse
!Point and click on the US capital.
? <M:p1>Paris<M:p2>
? <M:w1>Washington<M:w2>
? <M:l1>Lima<M:l2>
+<MOUSE w1 w2>;Very Good!
-<MOUSE p1 p2>;Paris is the capital of France.
-<MOUSE l1 l2>;Lima is the capital of Peru.
```

The <MARK> command is used to define the area in the Interactive (Question) Window where the student can point and click using the mouse to answer a question. The special response <MOUSE x y> is used to define how these answers will be judged and what action will follow them. The two “marks” specified by means of the <MARK> command define the upper left-hand and lower right-hand corners of the region in which a student may point and click to give an answer. This may be a single character, a line or a much larger area. When the student moves the cursor over an area which has been marked in the manner described above the cursor changes from an arrow to a pointing hand.

One Special Response OR Another

<OR>

no abbreviation

The <OR> Special Response operator is a special kind of Special Response. It was added in order to make authoring even easier. In some instances you may want WinCALIS to respond exactly the same way to two or more events, such as the click of the mouse on a certain hot link *or* the press of a certain key. You can use the <OR> Special Response to anticipate these two events in a single answer by linking two Special Responses with <OR>.

Interactive (Question) Window:
Click <M:s1>Start<M:s2> or press 's' to begin this session.

Correct/Continue Anticipated Response (Short Answer format tasks):

<MOUSE:s1 s2><OR><KEY 's'>

Both clicking the word “Start” in the Interactive (Question) Window or pressing “s” on the keyboard will cause WinCALIS to continue to the next task, since the responses are marked as Correct/Continue in the Short Answer dialog box.

The <OR> Special Response can be used to link more than two Special Responses, but it may not be used to link text responses with Special Responses.

The Pass Keys

<PASS>

<PASSBACK>

no abbreviation

Students who complete worksheets or take tests do not, frequently, start with the first item and work sequentially through the assignment. WinCALIS allows students to approach their lessons in exactly the same manner. The Pass keys (the gray + and the - key on the numerical keypad of the keyboard or the Ctrl+PageDown and Ctrl+PageUp keys) allow students to “pass by” a particular task that requires a response from them and return to it later.

Note: As a convenience for laptop users, WinCALIS now also supports the additional Pass Forward key Ctrl+PageDown and the additional Pass Back key Ctrl+PageUp.

The Pass Forward key moves on to the next task; the Pass Back key goes back to the closest previous uncompleted task. The primary task list may be thought of as a circle, for purposes of passing over tasks. Passing forward from the last task in the list will take a student to the first task, and passing backward from the first task will move to the last task. Sometimes, however, you may want WinCALIS to do more when the student passes on a question. For example, you may want the screen rewritten in some way before going on to the next question. In cases such as this the author may modify the Pass keys with the <PASS> special response.

#pass

/ <BO> ___ am happy.<BO>

? What pronoun completes the above sentence?

+I; Very good. This is correct.

&<T><BLA> I am happy.

=<PASS>;<T><BLA> ___ am happy.

#moveon

?<BLA> Now for the next step...

In the above example the Presentation (Text) Window is rewritten when the student passes. The author has decided to rewrite the sentence in the Presentation (Text) Window after the student's response. It originally appears with the sentence bolded. After passing the question the student moves on to the next task. The presence of the <PASS> special response triggers a specific action by WinCALIS. In this case, the sentence in the Presentation (Text) Window has been rewritten so that it is no longer bolded.

The <PASSBACK> special response is identical to <PASS>, but only goes into effect when the student passes backwards by pressing the Pass Back key. It is needed only in cases where a pass to a previous task must be differentiated from a pass to a subsequent task.

Notice that <PASS> still responds to either a pass to the next task or a pass to the previous task, <PASSBACK> must precede <PASS> in a list of anticipated responses. For example:

-<PASSBACK>; The student will see this feedback after
& passing back.
+<PASS>; The student will see this feedback after
& passing forward.

In the above example, student feedback is changed depending on which pass key is used. However, in the next example feedback does not change due to placement of <PASS> before <PASSBACK>.

+<PASS>; The student will see this feedback after
& passing either forward or back.
-<PASS BACK>; The student will never see this
& feedback.

Task Options and Defaults

Within a single script, different tasks need to function according to different rules. Some tasks ask students questions; these are recorded in the final score for the lesson. Other tasks simply present information, or allow a student to make selections. These should not be counted in the final score. Similarly, some tasks may need to be repeated several times, while others, having been successfully completed once, should not be.

Task options are the mechanism WinCALIS provides to modify the behavior of certain tasks. These options can be divided into three categories:

- Task options which deal with the way the task interacts with the entire lessons: can it be repeated?, can it be passed over?, should it count in the final score?
- Task options which relate to how the task will look to the student: should the previous contents of the Interactive (Question) Window be erased and this question given a clean screen?, should feedback to student responses be sent automatically to the D window? should the student who “gives up” have the opportunity to view the Correct/Continue answer?
- Task options which affect how student answers will be evaluated: should an answer otherwise correct be rejected for lack of proper capitalization, accented characters or punctuation?

Virtually all task options can be inserted automatically in WinCALIS Author, in the Task Options and Script Options dialog boxes.

Task options should not be confused with the commands described in the entire section “WinCALIS Commands.” This is particularly important since task options, like commands, are key words surrounded by angle brackets (e.g. <CAPITAL>), and two options (<BLANK> and <D>) have the same name as commands which perform similar functions. While WinCALIS commands may only appear in text that is sent to some window, task options may never occur there.

Note: All CALIS commands, options, and special responses are composed of one or more keywords in capital letters, surrounded by opening and closing angle brackets (<>), and separated by one or more delimiters.

Task options may only occur in two statements: a task label statement (*#tasklabel* <TASK OPTION> in which case the option holds for the specific task only), and default statements (@<TASK OPTION> where the option becomes the default for all subsequent tasks).

In the following example

```
@<NOSCORE><NOPASS><NOACCENT>  
#sample<ACCENT><NOPUNCT>
```

the default statement (@) instructs WinCALIS that subsequent tasks should not count in the final student score (<NOSCORE>), should not allow students to pass (<NOPASS>), and should not check whether a character is correctly accented (<NOACCENT>) to determine acceptable answers. All tasks following this statement will share these options until a subsequent default statement explicitly changes one of the three options. The task label (#sample) overrides one of the defaults (checking on correct use of accented characters, <ACCENT>), but does so only for the task #sample. Similarly, it relaxes the evaluation of punctuation (<NOPUNCT>) for the #sample task.

Task options affect flags that mark requirements for a particular task. All task options have two forms, one that turns the flag on and one that turns it off (e.g. <SCORE> and <NOSCORE>). In most cases the negative is simply the positive prefixed by a “NO.” WinCALIS begins each script with a set of implicit default behaviors. Task options need only be used when an author wants to bypass these implicit behaviors, or to change explicit modifications to the defaults made earlier in the same script.

Specifying Task-Lesson Interaction

Three task options specify the way tasks interact with the entire lesson. These are:

<PASS>/<NOPASS>
<REPEAT>/<NOREPEAT>
<SCORE>/<NOSCORE>

Note: Commands may never occur in either default or task label statements.

Allowing Students to “Pass Over” a Task

<PASS>
<NOPASS>

Abbreviation: <PAS>\<NOPAS>
WinCALIS default task option: <PASS>

The <PASS> task option allows students to pass over a particular task and come back to it later. The <NOPASS> option requires students to complete the task (or give up) before moving on. At the beginning of each lesson the default option of <PASS> is in effect. This should be changed only in unusual circumstances, when the lesson requires rigid enforcement of a sequential approach.

Allowing Repetition of Completed or Deleted Tasks

<REPEAT>
<NOREPEAT>

Abbreviation: <REP>\<NOR>
WinCALIS default task option: <NOREPEAT>

In general, tasks that ask a question and score students' answers should not be repeated in a single lesson, there is little point in asking a student a question already answered correctly. It may be useful to allow tasks that present information, or allow students to make choices, to be repeated several times. The <REPEAT> and <NOREPEAT> task options provide the author a tool to control whether or not a task should be repeated.

By default, when a student completes a task correctly or gives up, the task is marked (incorrect in the case of a give-up) and placed on a list of tasks which have been completed. Should that task appear later through the <INSERT> or <ADD> commands, WinCALIS will find that the task has been completed, and the <INSERT> or <ADD> will be ignored. The <REPEAT> option instructs WinCALIS not to check the list of tasks completed when inserting or adding this task to the student's list. The <REPEAT> option does not require that the task be repeated. It only allows for repetition of it if it is inserted or added to the task list more than once.

Counting Tasks in the Final Student Score

<SCORE>

<NOSCORE>

Abbreviation: <SCO> \ <NOSC>

WinCALIS default task option: <SCORE>

In most scripts, the majority of tasks will ask students questions, evaluate their answers, and compute a weighted score as the number of questions answered correctly out of the total number of questions. The weighted score is determined by totaling the following: a count of 1 for all tasks answered correctly on the first attempt; 1/2 for those answered correctly on the second attempt, 1/3 for those on the third attempt, and no points for attempts thereafter (see Figure 18 for view of score as shown to student in

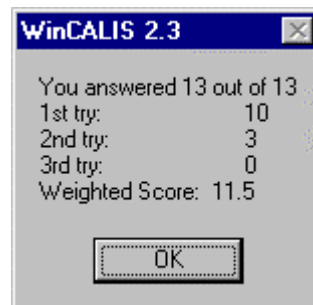


Figure 18: Weighted Score Report in WinCALIS

WinCALIS). An author may not want to count certain tasks which present tutorial information, provide remedial drill, or present a menu of choices for students to make. These tasks will not count for or against the student's final score if they contain the <NOSCORE> option.

By default, when a WinCALIS lesson starts, all tasks will count in the final student score. The exception to this rule is any task which contains no Correct/Continue answer (+) statements. When a task contains no Correct/Continue answer statements, tutorial, text, and question statements (if they occur) will be executed, and WinCALIS will mark the task completed without increasing the student's correct score or the number of questions attempted. Tasks which do not expect student responses do not utilize the <SCORE> option.

Modifying the Appearance of Tasks

Seven options allow authors to modify how WinCALIS presents a task to students. These are:

<BLANK> / <NOBLANK>

<D> / <NOD>

<PRAISE> / <NOPRAISE>

<SHOW>/<NOSHOW>
<LENGTH>/<NOLENGTH>
<KEY>/<NOKEY>
<MOUSE>/<NOMOUSE>

Clearing the Interactive (Question) Window for Each Task

<BLANK>
<NOBLANK>

Abbreviation: <BLA>\<NOBL>
WinCALIS default task option: <BLANK>

The WinCALIS default <BLANK> option causes the Interactive (Question) Window to be erased after the student attempts to answer the question, and to move the cursor back to row 1, column 1 of the Interactive (Question) Window before filling it in again. This requires the author to anticipate that students will not be able to view the mistakes they made on the first attempt during the second attempt at a task. The author may employ the <LAST> command to display the last student response on screen. (See the section “Advanced Screen Design Commands”) The <BLANK> task option affects all windows. You may remove these options using under the Script and Task Options menus, or by inserting the <NOBLANK> command in the appropriate window.

Using the <NOBLANK> option in the Presentation (Text) Window will cause the text lines of a new task to be added at the bottom of the previous Presentation (Text) Window. If a student gives a wrong answer, the same text will appear for a second time beneath the previous selection. Eventually the Presentation (Text) Window is filled and the text at the top is scrolled off of the screen to make room for new text.

Sending Feedback to the D Window

<D>
<NOD>

no abbreviation
WinCALIS default task option: <D>

WinCALIS responds to student answers in one of two ways; it can deliver feedback to the student regarding a Correct/Continue or Incorrect/Remain Here answer. In other lessons it can take on more complex tasks such as updating Text, Question, and/or Tutorial windows. The <D> option automatically sends feedback to student responses to the Feedback Diagnostic Window. The <NOD> option leaves the response text in the Interactive (Question) Window for tasks that need to update the Interactive (Question) Window before (or instead of) presenting evaluation text to the student. At the beginning of each lesson the default option <D> is in effect. This option will almost never need to be changed by authors who want to use feedback to student responses for the sole purpose of giving evaluations, hints, and clues to students.

Displaying “Correct” and “Incorrect” Messages in the Diagnostic Window

<PRAISE>

<NOPRAISE>

Abbreviation: <PR>\<NOPR>

WinCALIS default task option: <PRAISE>

WinCALIS automatically displays the message “Correct” in a Feedback Diagnostic Window (or D Box) when a student answers a question correctly. It also displays the message “Incorrect” when a student answers incorrectly (whether or not the Incorrect/Remain Here answer was anticipated). While this behavior is generally appropriate, an author may want to turn off these 'praises' and 'blames' at certain times. The <PRAISE> and <NOPRAISE> options allow authors to control this WinCALIS behavior. <PRAISE>, the default option at the start of each lesson, enables the 'praise' and 'blame' messages. <NOPRAISE> disables them. When the <NOPRAISE> option is selected, no automatic response will appear at the bottom of the screen when students type their answers.

Showing Correct/Continue Responses if a Student Gives Up

<SHOW>

<NOSHOW>

Abbreviation: <SHO>\<NOSH>

By default WinCALIS will display the Correct/Continue answer to a question, if the student makes one attempt and then presses F6, the Giveup key. In special circumstances (e.g. testing) this may not be desirable. The <SHOW> and <NOSHOW> options allow an author control over whether correct answers should be shown when students give up.

When WinCALIS starts a lesson, the default <SHOW> is in effect. This means that, should the student give up on a task, the answer field from the first Correct/Continue answer statement will be displayed. For example:

```
#talk <SHO>
?Would you like some punch?
?Yes, I _____.
+would; Very good. This parallels the question.
+do; Yes, this is possible.
```

If the student gives up on this task, the first Correct/Continue Anticipated Response will be displayed as the correct answer in the D box. As <SHOW> is the initial default, it is only necessary to use the <SHOW> task option explicitly, as in the above example, if a <NOSHOW> task option appeared earlier. The special response <GIVEUP> overrides the first correct answer rule. Instead of showing the first Correct/Continue answer when the student gives up, WinCALIS shows the text which follows the <GIVEUP> command. (Don't forget the ampersand.) <GIVEUP> also overrides the <SHOW>/<NOSHOW> option.

```
#talk <NOSH>
?Would you like some punch?
?Yes, I _____.
+{would|do}; Very good. This is a possible answer.
-<GIVEUP>; Correct Answer: would
```

WinCALIS will display “Correct Answer: would” when a student gives up, even though it has the <NOSHOW> option. The <GIVEUP> special response is more powerful than the <SHOW> option, since it may span more than one line, and may use all available commands.

Note: The Giveup response is generated automatically by WinCALIS Author.

Limiting the LENGTH of a Student's Answer

< LENGTH>

<NOLENGTH>

Abbreviation: <LEN>\<NOLEN>

WinCALIS default task option: <NOLENGTH>

This task option sets a limit on the amount of characters a student can type on the screen. When the student types more characters than necessary, WinCALIS beeps and shows the message “Your answer is too long.” The acceptable length is determined by WinCALIS to be the longest possible anticipated Correct/Continue response (those on a line starting with a +) given in that task. Wild cards are taken into consideration; however, the wild cards * and &, being of variable length, cannot be calculated and are ignored. (This means that the <LENGTH> option is not likely to be practical if these particular wild cards are used in anticipated Correct/Continue responses.) The amount of characters accepted can be set to a constant value with the <SET: ANSWER n> command (See the section “Setting Certain Numeric Values”). Consider this example:

@<LEN>

#1

? Where is the capital city of the U.S. located?

?

+Washington; That's right!

+{DC|D.C.}|Washington{,D.C.,DC}|{[Tt]he} Dist{.|ric}t of Columbia};

&That's right!

In the sample task above, the maximum answer can be 24 characters long, equalling the anticipated Correct/Continue response “The District of Columbia.”

<LENGTH:n>

Abbreviation: <LEN>

Limits the number of characters student may type to n characters. The command <LENGTH:0> is the same as <LENGTH>. {<LEN>}

Designating Mouse or KEYboard Input

<KEY>

<NOKEY>

no abbreviation

WinCALIS default task option: <KEY>

WinCALIS allows authors to disable keyboard input in cases where the student response should be given only by using the mouse or through the use of the <KEY> special response. This is possible by means of the <NOKEY> option. The WinCALIS default task option is <KEY>.

The keyboard is the most common source of student input. In some cases, when the mouse is used exclusively or where keystrokes may disrupt a screen, the author may wish to disable the student keyboard. The <NOKEY> option is used to disable student keyboard input.

When the <NOKEY> option is in effect, there is no blinking caret at the insertion point. Thus for multiple choice or True/False questions, which require only <KEY> or <MOUSE> special responses, use of the <NOKEY> option will “turn off the cursor.”

```
@<NOKEY>
#sample
!Use the mouse to click on
!the words marked in blue.
!You will not be able to use
!the keyboard.
?Click this <BLUE><M:m1>word<M:m2>.
+<MOUSE m1 m2>;
```

The student may only complete the task above using the mouse or a special response key. The option to produce on-screen characters is disabled.

The <KEY> task option is used to restore WinCALIS to its default behavior, i.e. to allow the student to type characters which are displayed on-screen. For example:

```
#sample<KEY>
?Type in the name of the U.S. president buried in Grant's tomb.
+Grant; Good!
```

The student may use the keyboard to type in a response to a question.

Disabling the Mouse Hand Pointer

<MOUSE>

<NOMOUSE>

no abbreviation
WinCALIS default task option: <MOUSE>

The mouse pointer (mouse cursor) in WinCALIS is normally an arrow. It turns into a hand pointer when it is over a “hotspot” hyperlink on the screen. WinCALIS allows authors to disable the mouse hand pointer in cases where the hand pointer is undesirable, such as when the student is asked to click on a single Correct/Continue hotspot in text or in a graphic on the screen. This is possible by means of the <NOMOUSE> option. The WinCALIS default task option is <MOUSE>.

WinCALIS Answer Evaluation Options

WinCALIS provides four task options that control how student answers should be evaluated. These are:

```
<ACCENT>/<NOACCENT>
<ALL>/<ANY>
<CAPITAL>/<NOCAPITAL>
<PUNCT>/<NOPUNCT>
```

Requiring Correct Placement of Accents

<ACCENT>

<NOACCENT>

Abbreviation: <ACC>\<NOA>

WinCALIS default task option: <ACCENT>

For languages that use accents and other diacritical marks, authors may wish to insist on correct use of accented characters. The <ACCENT> and <NOACCENT> task options allow the author to specify if a student answer that matches the anticipated answer in all respects except for accents (e.g. the same vowels and consonants appear but without diacritical marks, or with incorrect diacritical marks), should be counted correct. <ACCENT> requires proper use of accented characters, <NOACCENT> relaxes this criterion.

This option is helpful in languages, such as English and Russian, which have no fully consistent stress pattern. Stress marks do not appear in the standard written language; however, they are often used in language instruction.

At the beginning of each lesson the default option <ACCENT> is in effect. An author need not specify the <ACCENT> option unless the default is explicitly changed.

Note: <ACCENT> and <NOACCENT> do not apply only to Correct/Continue answers. These options apply also to anticipated Incorrect/Remain Here (-) and neutral (=, \$) responses. Do not select the <NOACCENT> option and then create a statement that anticipates the absence or incorrect use of an accented character.

Requiring Students to Match One or All Correct/Continue Answer Possibilities

<ANY>

<ALL>

no abbreviation

WinCALIS default task option: <ANY>

Correct/Continue answer statements may function within a task in one of two ways. They may indicate a number of Correct/Continue answers, any one of which is acceptable. The student who types any one Correct/Continue answer completes the task successfully. Correct/Continue answers may also function as separate parts of an entire task. In this case students must match each part of the whole, one answer at a time, until all Correct/Continue answers have been matched. The second option functions like one large question with multiple parts.

The <ANY> task option instructs WinCALIS to accept any Correct/Continue student answer and mark the task complete. The <ALL> task option requires that students match each Correct/Continue answer, one at a time.

#sample <ANY>

/ Look at the following paragraph and find one

/ reference to energy.

/ American nuclear electricity was born in 1956.

/ By 1973, it had become a technology America

/ could turn to when faced with the oil crisis.

/

? What is one reference to energy?
+nuclear;Yes, “nuclear” refers to the way energy
&is generated.
+electricity;Good. Electricity is a type of energy.
+oil;Correct. Oil is an important source of energy.

In the above example, the task is successfully completed if student answers “nuclear,” “electricity,” or “oil.”

#ex <ALL>
/ Look at the following paragraph and find all
/ references to energy.
/
/ American nuclear electricity was born in 1956.
/ By 1973 it had become a technology America
/could turn to when faced with the oil crisis.
/
? What is one reference to energy?
+nuclear;Yes, “nuclear” refers to the way energy
&is generated
+electricity;Good. Electricity is a type of energy.
+oil;Correct. Oil is an important source of energy.

In this latter sample task, a student must answer “nuclear,” “electricity,” and “oil” before completing the task successfully. The <ALL> option requires students to find all three words, one at a time. Students may type the Correct/Continue answers in any order. When the last answer is given, the student advances to the next task.

When a student gives up with the <ALL> task option in effect the response WinCALIS gives is slightly different than when it is not in effect. Instead of replying “Correct Answer:” and listing the first anticipated Correct/Continue response, WinCALIS replies: “Remaining Answers:” and lists all those Correct/Continue responses that the student was unable to come up with.

It should also be noted that the <ALL> task option command can be used to require Correct/Continue responses of different types, involving both special responses and regular keyboard input. For example:

#sample_all<ALL><BLA>
/ First CLICK the LAST name of the president
/written about in this piece of text.
/Next TYPE his FIRST name fully.
/
/<M:fn>_____ <M:ln>_____, the 16th president
/of the United States, guided the country through
/the most devastating experience in its national
/history, the CIVIL WAR. He is considered by
/many historians to have been
/<TAB:2>the greatest American president.

#sample2<ALL><BLA>
?
?<TAB:3><M:a1>Lincoln<M:a2>
?<TAB:3><M:b1>Washington<M:b2>
?<TAB:3><M:c1>Carter<M:c2>
?
?<CE><M:prompt> <CUR:prompt>

```
+Abraham;<T><CUR:fn>Abraham<M:ln><Q> <CUR:prompt>
+<MOUSE a1 a2>;<T><CUR:ln>Lincoln<Q> <CUR:prompt>
<MOUSE b1 b2>;Incorrect
<MOUSE c1 c2>;Incorrect
&;Incorrect
```

In the #sample2 task, in which a special response (<MOUSE a1 a2>) and a regular keyboard input response “Abraham” have both been defined as Correct/Continue Anticipated Responses, the <ALL> command will require the student to both type and click the Correct/Continue answers, i.e. <M:a1>Lincoln<M:a2> and “Abraham,” before WinCALIS responds with feedback.

<ALL: n>

Parameter: (n) the number of correct answers a student is required to give
WinCALIS default task option: <ANY>

If there is a lengthy set of Correct/Continue answers and the student must type more than one of the Correct/Continue answers, but not every Correct/Continue answer, you can use the <ALL:n> version of the <ALL> option. The number of Correct/Continue answers is indicated by n.

The All n Answers option requires the student to match more than one Correct/Continue answer, one at a time, but fewer than the total number of Correct/Continue answers.

Requiring Correct Capitalization

<CAPITAL>

<NOCAPITAL>

Abbreviation: <CAP>\<NOCAP>

WinCALIS default task option: <CAPITAL>

In some languages the capitalization of certain words is very important (e.g., nouns in German): and in others it is not a part of the original language (e.g. classical Greek). The <CAPITAL> and <NOCAPITAL> task options allow the author to specify if a student answer that matches the anticipated answer except for capitalization should be counted correct. <CAPITAL> requires proper use of capitals, <NOCAPITAL> relaxes this standard.

Note: The options <CAPITAL> and <NOCAPITAL> do not only apply to Correct/Continue answers. These options also apply to anticipated Incorrect/Remain Here (-) and neutral (=, \$) responses. Do not select the <NOCAPITAL> option and then create a statement that anticipates the lack of a capital letter.

Requiring Correct Placement of Punctuation

<PUNCT>

<NOPUNCT>

Abbreviation: <PUN>\<NOPU>

WinCALIS default task option: <PUNCT>

By default WinCALIS requires the student answer to contain all punctuation listed in the Correct/Continue response. Missing or extraneous punctuation will be considered incorrect. The <PUNCT> and <NOPUNCT> task options allow the author to specify if a student answer that matches the anticipated answer except for the use of punctuation should be counted correct. <PUNCT> requires proper use of punctuation. <NOPUNCT> relaxes this standard.

Note: <PUNCT> and <NOPUNCT> do not only apply to Correct/Continue answers. These options also apply to anticipated Incorrect/Remain Here (-) and neutral (=, \$) responses. Do not select the <NOPUNCT> option and then create a statement that anticipates the absence or misuse of punctuation.

Permitting Students to Give Up on Tasks

<GIVEUP>

<NOGIVEUP>

Abbreviation: <GIVE>\<NOGIVE>

WinCALIS default task option: <GIVEUP>

The special response <GIVEUP> is also a task option default. Students are permitted to give up on a task; whether or not they see the Correct/Continue answer depends on the other task options in effect. This option is removed by use of the <NOGIVEUP> task option which requires the student to enter the Correct/Continue answer in order to finish the lesson. A student unable to give the desired answer would be forced to leave the exercise using the Esc key. Only in unusual circumstances is this task option useful.

Setting the Minimum Number of Attempts before Giving Up

<GIVE: n>

Parameter: (n) the minimum number of tries before a student is allowed to give up

WinCALIS default task option: <GIVE: 1>

#sample<GIVE:3>

?What city is the capital of Albania?

+Tirana

In the above example, the student must try to answer the question at least three times before the WinCALIS Giveup key (F6) is enabled and the Correct/Continue answer can be shown. If the <GIVE: n> task option is used as a global default, then resetting it with the value 1: <GIVE:1> will restore WinCALIS to its default setting. The student may hit the Giveup key (F6) after one attempt has been made to answer the question and see the Correct/Continue answer. Giving up on a task causes it to be scored as incorrect.

Limiting the Number of Attempts

<MAXTRIES: n>

no abbreviation

WinCALIS default task option: <MAXTRIES: 99>

The <MAXTRIES: n> task option limits the number of attempts a student can make at a task. By default, a student may attempt to give the Correct/Continue answer until successful or to hit the Giveup key (F6) and be shown the Correct/Continue answer. The task option <MAXTRIES: 3> will automatically show the student the message “Maximum number of tries reached” and will count the task as incorrect. This task option is useful for WinCALIS exercises which are scored tests.

Group Behavior

<GROUP:grouptask [,exitgrouptask]>

no abbreviation

Parameter: (grouptask) the label of the entry group task; (exitgrouptask) the label of the exit group task

WinCALIS default task option: <GROUP: 1>

Identifies group tasks to be executed upon entry to and exit from a task group.

The addition of “group structure” to WinCALIS is principally motivated by the goal of displaying the proper directions to the student when he enters a new section of the script, either by working tasks or browsing, either forwards or backwards, in either linear or random order, and also whenever he requests them with the F9 key. This has never before been possible, including in Text CALIS. Previously, if a student passed backward into a task, he could easily never see the proper directions for that task. This inconvenience has now been eliminated.

A WinCALIS script can now be segmented into blocks, or groups, each of which can have its own entry “group task” and exit “group task.” These tasks are executed, if present, whenever the group is entered or exited.

Each group is added by WinCALIS Author to the secondary task list with its own <ADD> command. See the example script below.

The group task takes its task label in WinCALIS Author from the first task in the group. For example, if directions are written in the Popup (Directions) Window for Task #1, then WinCALIS Author will automatically generate a group task labeled #group1, which will contain those directions. Task #1 will have the task option <GROUP: group1> written to its task. Every task will belong to the same group and share group task #group1, until new directions are written in the Directions Window.

A group task should always have the task options <NOSC> and <REP>.

WinCALIS Author automatically generates a group task in the following situations:

- when the Popup (Directions) Window has something written in it (entry group task)
- to draw the screen at the beginning of a cloze (entry group task)
- to allow the student to savor his completed work before continuing at the end of a cloze (exit group task)

Authors can make use of the new <GROUP> option themselves for structuring long lessons (e.g., an hour-long mid-term test) into self-contained units, and in many other unexplored ways.

A Sample Script With Group Tasks for Directions and Cloze

In the script below, there are four separate task groups, tasks 1-2, task 3, tasks 4.1-4.4 (a cloze), and task 5. Each group is added by WinCALIS Author to the secondary task list with its own <ADD> command in a separate statement in the #0 task. Each group has an entry group task generated by WinCALIS Author, containing the directions for that group. The group task has a task label of the form #groupn, where 'n' is the number of the first task in the group (the first digit of the task number for a cloze, numbered 1.1, 1.2, 1.4, etc.): #group1, #group3, #group4, and #group5. In addition, the cloze 4.1-4.4 has an exit group task, #continue4, which is automatically generated by WinCALIS Author for cloze tasks, so that WinCALIS will pause before blanking the screen, after the student fills in the last

blank in the cloze. Notice also that the <GROUPLOCK> option has automatically been added to each of the cloze subtasks, to ensure integrity of the screen while working on the cloze, by preventing the student from passing out of the cloze. The group task #group5 is a empty dummy task, the purpose of which is to define task #5 (and any other tasks which happen to follow it) as a group.

Parts of the script not relevant to the group structure have been omitted below. To see the full script, which is included as a sample exercise in the Author Workstation Setup ("English" subdirectory), see the files "accompst.wca" and "accompst.cal."

```
#0<NOSC><NOPR>
?<ADD: 1--2>
?<ADD: 3--3>
?<ADD: 4.1--4.4>
?<ADD: 5--5>
#end
```

```
#group1<NOSC><REP>
!<CE> Please answer each question based on the text.
!<CE>Pay careful attention to the vocabulary in boldface.
```

```
#1<GROUP:group1>
/There seems to be a trend in French cinema today to make films about classical
/musicians: <IT>Tous Les Matins Du Monde<IT>, <IT>Un Coeur en Hiver<IT> and ...
?This excerpt would be best described as:
?<TAB:5><M:a1><UL>a.<UL> an article<M:a2>
-<MOUSE:Q a1 a2><OR><KEY: 'a'>
?<TAB:5><M:b1><UL>b.<UL> an advice column<M:b2>
-<MOUSE:Q b1 b2><OR><KEY: 'b'>
?<TAB:5><M:c1><UL>c.<UL> a film review<M:c2>
+<MOUSE:Q c1 c2><OR><KEY: 'c'>
```

```
#2<GROUP:group1>
?The emphasis of the film is on the political strife
?of Nazi-occupied Paris during World War II.
?<CE><M:t1><UL>T<UL> rue<M:t2>      <M:ff1><UL>F<UL> else<M:ff2>
+<MOUSE:ff1 ff2> <OR> <KEY: 'f'>;
-<MOUSE:t1 t2> <OR> <KEY: 't'>
```

```
#group3<NOSC><REP>
!<CE>The following question is a short answer.
!<CE>If you have questions, pressing SHIFT + F2 may help!
#3<GROUP:group3>
?<CE>Briefly describe Sophie.
+Sample answer: She is a 20-year old pianist
+& {20|twenty|musician|pianist|play*|piano|[Aa]companionist|assistant}&
-&;<CE>If you are having trouble, please press SHIFT+F2.
```

```
#group4<NOSC><REP>
!<CE>In the next part you will need to fill in the blanks, based on the context.
!<CE>You can press Shift + "?" or click on the Clue button to see or hear a clue.
/<BLA>
?<BLA>
/<M:a1>World War II<M:a1b>      <M:a2>Charles<M:a2b>      <M:P1>The Accompanist<M:P1b>
<M:P4>Sophie<M:P4b>      <M:b1>relationship<M:b1b>      <M:P2>friendship<M:P2b>
<M:P3>Irene Brice<M:P3b>
```

?
 ? <M:ANS_1>_____<M:ANS_1b> chronicles a
 <M:ANS_2>_____<M:ANS_2b> between
 ? an opera star, <M:ANS_3>_____<M:ANS_3b>, and her <M:CLUE_1>pianist,
 <M:ANS_4>_____<M:ANS_4b>.
 ?
 ?<M:BOTTOM>

#4.1<LEN:15><GROUP:group4,continue4><GROUPLOCK>
 ?<CUR:ANS_1>_____<CUR:ANS_1>
 =<MOUSE ANS_2 ANS_2b>;<GOTO 4.2>
 =<MOUSE ANS_3 ANS_3b>;<GOTO 4.3>
 =<MOUSE ANS_4 ANS_4b>;<GOTO 4.4>
 +The Accompanist;<Q><CUR:ANS_1> _____<CUR:ANS_1><M:ANS_1b><BLUE><BO>The
 Accompanist<BLUE><BO><M:ANS_1>
 &<T><CUR:P1><FILL:15 \><CUR:P1><M:P1b><D>
 +<MOUSE P1 P1b>;<Q><CUR:ANS_1> _____<CUR:ANS_1><M:ANS_1b><BLUE><BO>The
 Accompanist<BLUE><BO><M:ANS_1>
 &<T><CUR:P1><FILL:15 \><CUR:P1><M:P1b><D>

#4.2<LEN:15><GROUP:group4,continue4><GROUPLOCK>
 ?<CUR:ANS_2>_____<CUR:ANS_2>
 =<MOUSE ANS_1 ANS_1b>;<GOTO 4.1>
 =<MOUSE ANS_3 ANS_3b>;<GOTO 4.3>
 =<MOUSE ANS_4 ANS_4b>;<GOTO 4.4>
 +friendship;<Q><CUR:ANS_2> _____<CUR:ANS_2>
 <M:ANS_2b><BLUE><BO>friendship<BLUE><BO><M:ANS_2>
 &<T><CUR:P2><FILL:10 \><CUR:P2><M:P2b><D>
 +<MOUSE P2 P2b>;<Q><CUR:ANS_2> _____<CUR:ANS_2>
 <M:ANS_2b><BLUE><BO>friendship<BLUE><BO><M:ANS_2>
 &<T><CUR:P2><FILL:10 \><CUR:P2><M:P2b><D>

#4.3<LEN:15><GROUP:group4,continue4><GROUPLOCK>

...

#4.4<LEN:15><GROUP:group4,continue4><GROUPLOCK>

...

#continue4<LEN:1><REP><NOPR><NOPASS><NOSC>
 ?<CUR:BOTTOM><CE><M:C>Press Enter to continue<M:C2>
 +<ENTER>;
 +<MOUSE C C2>;
 +<KEY 'e>;
 +&;

#group5<NOSC><REP>

#5<GROUP:group5><ALL: 6><NOCAP>
 /<BLA>There seems to be a trend in French cinema today to make films about ...
 ?In a review of a film about personal relations, you would expect to find
 ?specific words which deal with relations. Find six such words. Type your
 ?answers one at a time.
 +friendship;<CE>Yes, that's one!
 +accompanist;<CE>Yes, that's one!

<GROUPLOCK>

no abbreviation

The <GROUPLOCK> task option prevents the student from passing out of the current group until all tasks in the group are completed.

This option is a companion to the <GROUP> option. A check box in the Script Options, More Options and Task Options, More Options dialog boxes in WinCALIS Author allows you to turn on and off the <GROUPLOCK> option.

However, <GROUPLOCK> is turned on automatically by WinCALIS Author for a cloze group, since the cloze text, other than the blanks to be filled in, is written to the screen only once in the group task, and must not be erased or overwritten.

Note that without special arrangements in the script to prevent it, <GROUPLOCK> can produce the “trapped student” effect, also known as the “Hotel California” problem. While browsing, the student can pass into a group with the <GROUPLOCK> option, but once there, he cannot leave, until he has completed all the tasks in the group.

Turning an Exercise into a Quickie Quiz

<QUIZ>

no abbreviation

Selecting the <QUIZ> script option turns on “Quiz” mode. This requires the student to log in at the beginning of a lesson, supplying his name and identification number (in the U.S. his social security number). WinCALIS then records the student's responses in a disk file in the current directory with the filename “nameidno.scr.”

Your Key to Creativity

You now have the knowledge to empower your lessons in WinCALIS Author with enhancements using CALIScript itself. Remember, it is not necessary to go outside of WinCALIS Author in order to make interesting and challenging scripts. However, with some knowledge of CALIS commands you will begin to realize the full potential of WinCALIS as an authoring tool.

Chapter 6: Multimedia in WinCALIS

Overview of Multimedia

Multimedia Requirements

Hardware

In order to implement audio, video or animation into WinCALIS lessons, you must have first installed the proper hardware and software. For installation of a SoundBlaster card or other MCI-compliant cards, please refer to the manuals provided by the manufacturer.

Software

WinCALIS supports a virtually unlimited range of multimedia through the Windows Media Control Interface (MCI). In order to operate your hardware with WinCALIS Multimedia Editor, you must have the MCI software drivers installed. It is easy to check if your hardware is properly installed for MCI while in Windows.

For the convenience of WinCALIS Authors and system administrators, Disk 4 of the WinCALIS Author Workstation Setup and Disk 2 of the WinCALIS Student Workstation Setup installation disks include MCI drivers in the subdirectory "Drivers" for several common laserdisc players or VCRs, since the installation of these players at many institutions may predate the establishment of the MCI standard. In addition, there are MCI drivers for the run-time players of three popular animation packages used by WinCALIS authors: Microsoft Multimedia Movie Player, Autodesk Animation Player, and Gold Disk Animation Works. For how to install these drivers, see the section "Installing MCI Drivers" in the chapter on "Configuring and Customizing WinCALIS."

Note: WinCALIS no longer supports any hardware through WinCALIS 1.0's proprietary drivers)

To check hardware for MCI:

- Go to the Program Manager screen and click on the Control Panel in the Main group.
- Click on the Drivers icon.
- Scroll through the list of drivers until you get to the [MCI] section. You should see a driver listed for your device if it is properly installed for MCI.

To check for MCI while in the Accessories menu:

- Click on the icon Media Player. A Media Player dialog box will appear.
- Select the menu item Devices.
- If your hardware or software device is listed, it is installed for MCI.

If your device does not appear in the list, contact the vendor from which you acquired the hardware or software, the technical support department of the manufacturer, or the Humanities Computing Laboratory.

Function of Multimedia

The use of audio, video, graphics, and animation can enhance the appeal and educational value of any lesson. As a teacher you can use various audio techniques to expose students to a wide variety of native accents, thereby deepening their knowledge of the language and cultures they are studying. You may incorporate music into lessons to test student's recognition and comprehension of various clips. Through video you can enrich the student's knowledge of cultures in which the language is utilized. Again, video can be used for informational purposes, or as a selection to ask questions about. Full motion video also gives you the option of displaying still photographs by playing a tape a single frame at a time. Graphics and animation can be incorporated into lessons to help illustrate concepts (for instance, the relationship between the prepositions "to" and "from"). These forms of visual media are more economical in space requirements and demands on the computer's video display capacity.

More Tips on Multimedia in WinCALIS

Background

"Multimedia" refers to audio and video enhancements to the pure text and still graphics on the computer screen--in other words, sound and moving pictures. Typical sources of sound are voice tape recordings played on a cassette recorder connected to an external plug on the computer, or the same voice digitized and saved as audio files stored on the computer's disk. "Moving pictures" can similarly come from videotapes and laserdiscs played on a video cassette recorder (VCR) or laserdisc player connected to an external plug on the computer, or the same images digitized and saved as full-motion video files on the computer's disk.

Analog vs. Digital

Note that in the case of both audio and video, the source can be an external "analog" peripheral device such as a tape recorder or laserdisc player, or it can be simply a digital file stored on disk. A significant trend of the 1990s is for digital files to replace external peripheral devices as the primary source of multimedia, because of the exponential increase in disk storage capacity and the ability of networked workstations to deliver information, including audio and video information, over long distances from central mass storage devices ("the information superhighway").

Note: Disk storage usually refers to a computer's hard disk drive or shared network drive, but also includes other digital storing devices such as local or remote CD-ROM drives, optical jukeboxes, and Bernoulli cartridges.

Digitized multimedia is cheaper, faster, and easier to install and maintain than the elaborate external devices required to play analog audio and video recordings. Audio and video tapes and laserdiscs are at the moment still the more practical way to deliver hours of audio and video, such as would be required for viewing a feature-length movie without interruption, but the typical Computer-Assisted Language Learning exercise employing multimedia rarely requires more than a few minutes of audio or video material in a single session, so multimedia in digital form is quite feasible for CALL. WinCALIS documentation here highlights this increasingly dominant form of multimedia.

MIDI and Audio CDs

Although for foreign language learning applications, voice recordings, and movies which include voice soundtracks are by far the most common sources of multimedia, the range of sound and moving pictures is much broader than these sources. Sound also can include music (or spoken voice) on audio CDs and MIDI (Musical Instrument Digital Interface) electronic music. Songs can be used

educationally for their lyrics. But music from audio CDs or MIDI could just as well be incorporated in exercises as background “mood music” to accompany the student as he or she works.

Animation

“Moving pictures” also includes a wide variety of animation. Animation provides the same visual appeal as live video (as Disney long ago discovered), while being more economical in space requirements and demands on the computer's video display capacity. Because of its stark simplicity, animation also can often illustrate a concept (e.g., the working of the internal combustion engine, the dynamic relation of the prepositions “to” and “from”) more effectively than live video can.

Photographic Images

Multimedia full-motion video also, somewhat paradoxically, adds the option of still photographs to the existing choice of bitmap graphics, by recording or playing back video a single frame at a time, without audio. The Kodak Photo CD standard on CD-ROM is an example of this. In the near future, a connection between the computer and now separate devices such as the telephone and the TV will enrich the variety of audio and video multimedia available to applications such as WinCALIS.

Designing a Multimedia Lesson

The first step in including multimedia in your WinCALIS script is to have a clear overall design plan, in which you know what you want the student to hear or see, and at what point in the exercise. A flow chart or detailed outline of the lesson is good for this. But it is also possible to make notes right in the script as to where video and audio belong, as you compose the script in WinCALIS Author. Media events can be inserted in the script both as a stimulus to the student, in any WinCALIS window--Question, Text, Directions, even Lesson Description (not recommended)--or as a reaction to an action on the student's part, by placing the media event in the “D” window as feedback to the student's response.

There are numerous other options for playing video, and to a lesser extent audio--such as which track to play among multiple tracks; where on the screen to place the video image, and in what size and shape; whether to play video with the soundtrack turned off, or conversely whether to play just the soundtrack; whether to crop the video image, focusing attention on just a part of it; and so on.

All such decisions about the timing and placement of a media event in the script are made in the WinCALIS Author, not in the Multimedia Editor. But Author does not allow hearing or viewing the media event itself. It is the function of the Multimedia Editor to permit this review of the media itself, together with determining when the clip for each event should begin and end.

No matter what, it is safest and most efficient to have the WinCALIS script fully or at least partially completed (at least a few tasks), in place and tested in WinCALIS, before using the Multimedia Editor to review and mark up the audio or video component of the script. Sometimes it is useful to make a small mockup or prototype of the full script, with just one or two tasks and a small sample of the desired media.

Preparing Multimedia Files

Before you can use WinCALIS Multimedia Editor to manipulate files for insertion into a WinCALIS script, you must make them readable by the Multimedia Editor. This process is different for audio, video, and animated media. Each is explained below under its respective heading.

Audio

Audio files are among the simplest to record and edit using the Multimedia Editor. To record sound files into the Multimedia Editor you must have a sound card installed. If you have access to digital audio files, you still need a sound card in order to produce sound from your machine. Your device must be properly installed for MCI in order for WinCALIS Multimedia Editor to read the audio *.WAV files.

Video

The Multimedia Editor does not create the initial video file. You must create the video file first using specific hardware and software. You may create a file in one of two ways. If you have an overlay card you may use and manipulate laserdisc players and VCRs in the Multimedia Editor. When using an overlay card, the Multimedia Editor appears to function in the same way as when directly recording an audio file. The difference is that the overlay card acts as an intermediary between the Multimedia Editor and the hardware, telling it when to start and stop. In order to display video using this method you would need to have an overlay card installed. An alternative method involves digitized video. In order to digitize video you need to have the digitizing hardware and software, that is-- something that will translate the analog signals coming from your VCR or laserdisc into digital signals which can be read and stored on your computer's hard disk. Once the video is digitized you do not need the hardware or software required to digitize it in order to view it. The advantage is that it can be viewed on any machine which has Video for Windows (included in the full WinCALIS Setup package.)

Note: For digital video, it is necessary to have a sound card installed in order to hear any sound accompanying your visual input.

Animation

Preparing animation for editing in Multimedia Editor is very simple. It is not necessary to have any hardware in order to create or play animation files, however, you do need the proper software. In order to play animation files on other machines (for instance in a WinCALIS lesson) you would need to have the animation driver installed, but not the software itself.

Recording Audio

There are at least three ways to record sound files. The first is to use the WinCALIS Multimedia Editor, the second is to use the Windows Sound Recorder (in the "Accessories" group), and the third is to use the audio editor included with your audio card, such as the Sound Blaster sound editor. The Windows Sound Recorder is extremely easy to use, but imposes a limit on the length of the audio you may record. We recommend that you use the WinCALIS Multimedia Editor, which is tailored to the needs of WinCALIS authors.

You will first record digital sound temporarily to a buffer in RAM memory, then you will save it as a permanent disk file, which can be copied, renamed, and deleted just like a text file. A Windows digital waveform audio file usually has the file extension *.WAV. The length of the file is limited only by the size of the storage device (usually your hard disk) that the file is being saved on. Keep in mind that a sound file can grow very quickly.

Recording audio is a simple and quick procedure. After you have launched the Multimedia Editor (See section entitled "WinCALIS Multimedia Editor") you will be presented with a window with various controls. You will notice that the screen has a microphone Record icon at the lower left. The Multimedia Editor is ready to record at any time. It is not necessary to select a device from the Device menu unless you wish to open a previously recorded *.WAV file.

Clicking on the Record icon will begin recording your voice. You will notice a green line displayed in the middle of the window in the upper part of the screen. When you play back from disk a previously recorded *.WAV file, the thickness of this line will vary with the amplitude of the audio signal, giving you a visual cue to aid in the marking process. To stop recording, click on the Stop button with the

black square. To continue recording, click again on the microphone Record icon. This will append the new recording to the end of the previous one. To start over fresh, click on "File, New." In order to listen to what you have recorded, click on the triangular Play icon. You can position the recording to any point with the tape recorder-style controls at the bottom, or by dragging the slider along the horizontal scroll bar with the mouse.

When you are satisfied with your recording, it is time to move to the second stage, in which you mark segments, or "clips," for WinCALIS to play. The second part of preparing an audio file involves playing your recording and marking clips that you may want to use in the script. This procedure is the same as for existing media sources. See the section Marking Clips in an Existing Multimedia Source on defining clips.

You may record a *.WAV file, then immediately mark clips, and then save the *.WAV file when prompted at the time you save the *.MMD clip file. This is the procedure which the design of the Multimedia Editor encourages. Or you may save the newly recorded wave file to disk before marking any clips, or in an entirely separate session, by clicking on the "File" menu and selecting "New" or "Exit," or by selecting a new device from the "Device" menu. In either case, be sure to move to the subdirectory which contains your script before saving your digital audio *.WAV file and *.MMD clip file.

The recording source in the typical installation is by default microphone input. If the source of your recording is a "line" source, such as the headphone jack or "line out" of a radio, TV, audio CD in a CD player or CD-ROM drive, or a tape recorder (playing a prerecorded audio tape), connected to the "line" jack on your audio card, you should launch the "Mixer" application which is usually installed as part of the setup of your audio card, or the Windows Volume Control, and select the correct source from which to record, at the same time deselecting the microphone.

To record an audio segment:

- Prepare your external device (i.e. connect the microphone or other external device)
- Click on the Record button at the bottom left of the Multimedia Window. Once you click this button the Multimedia Editor will begin recording.
- Record your material
- When finished recording, click on the Stop button.
- Save the file by going to the file manager and selecting Save. A Save dialog box will appear. Type the name you want to save the file under.
- Click on the OK button.

Note: Be sure to click on the Stop button when you finish recording or the WinCALIS Multimedia Editor will continue to record until you run out of hard disk storage.

When re-editing wave files, pressing Record to record new material only inserts the new material or appends it to the existing material. It does not erase and record over previously recorded material. If you wish to record over or perform more elaborate editing on an audio clip, use one of the many specialized audio editors available, such as the Windows Sound Recorder or the audio editor installed along with your sound card.

.WAV and *.MMD Files

When you save your recording, the file will automatically be given a .WAV extension. This is the standard extension that Windows recognizes for all audio files. In order to create audio clips that can be inserted into WinCALIS lessons, the Multimedia Editor creates another file, an *.MMD file. This file stores information about the individual clips that you produce from each recording, such as the starting and stopping points for each clip. The *.MMD files also act as a link between WinCALIS lessons and the various multimedia files. It is the *.MMD file that tells WinCALIS what clips can be used in WinCALIS lessons.

Creating *.MMD Files

Once you have material you have recorded or which has been previously created you are ready to create clips in the Multimedia Editor. The clips will all be saved under one *.MMD file. Any *.MMD file can have any number of clips from several different devices. The *.MMD file must have the same name as the lesson in which you want to incorporate the clips. WinCALIS Author will recognize only those files which have the filename of the lesson you are currently working on (although the extension is different.) So then, if you had a lesson entitled BEERCOM1.WCA, the *.CAL script would be generated as BEERCOM1.CAL and the *.MMD file would appear BEERCOM1.MMD. Because these files all have the same filename, WinCALIS would recognize them as belonging together.

WinCALIS Support for Multimedia

Audio

Microsoft Windows has built-in support for three kinds of audio media, given the proper hardware (See "Multimedia Requirements"):

- digital "waveform" audio (digital sound files with the extension .WAV)
- MIDI files, (via the "MIDI Sequencer" driver, usually with the file extension *.MID or *.RMI)
- CD Audio (music CDs played in the computer's CD-ROM drive) with the sound directed from the headphone jack to headphones or external powered speakers, or alternatively, from the headphone jack or an internal connector to the "line in" or "CD" connector on an internal audio card, and then to headphones or external speakers.

Video

WinCALIS will support any video product that is MCI-compatible, or has MCI-compliant drivers. The full Setup package includes an installation disk for Windows digital-full motion video software, Video for Windows 1.1, as well as MCI drivers for "ViSCA"-compatible VCRs and some common laserdisc players from Sony, Pioneer, and Panasonic:

- Pioneer 2200, Pioneer 4400
- Sony Lasermax
- Panasonic Videodisc Player

The setup disk for Video for Windows is included as a separate diskette in the WinCALIS package, and should be installed, if necessary, on each machine that will be used to show digital video files.

The MCI drivers for the laserdisc players and VCRs can be installed using Disk 4, "Multimedia" of the WinCALIS Setup program. See the section Installing MCI Drivers For Multimedia.

Note: A good quality and properly installed video display card is more important for digital video than a fast CPU processor speed. An audio card and speakers are required in order to hear the sound track. The optimal video mode for digital video is standard 640x480 VGA or any higher resolution *with 256-colors*.

Animation

WinCALIS includes MCI drivers for three popular animation packages used by WinCALIS. These drivers can be installed using Disk 4, "Multimedia" and are available for the following packages:

- Microsoft Multimedia Movie Player
- Autodesk Animation Player
- Gold Disk Animation Works

Graphics

Requirements

There are no special hardware or software requirements for including graphics in your WinCALIS lessons, since Windows already provides extensive graphics support. However, in order to create graphics from scratch or draw on existing clip art, you must have the proper software. Please consult the documentation for your software for further information. Furthermore, in order to understand how the graphics you include will appear on student workstations, you should have a basic understanding of the types of graphics available.

Creating Graphics

Although graphics are considered a form of multimedia, you do not use the Multimedia Editor when dealing with graphics. You may incorporate any graphic which carries a .BMP file extension into WinCALIS. Therefore, you can use any graphics software which creates *.BMP files. If the program does not support *.BMP files you may cut the image, paste it into Paintbrush in Windows, and save the file with a .BMP extension. If you create your graphic in Paintbrush you do not have to convert it, as it will automatically be given the .BMP extension.

Saving Graphics

In order for your graphic files to be available for incorporation into lessons they must be visible in the Graphic Dialog Box. You may save the graphic in one of two directories to make it available to WinCALIS:

- In the WinCALIS Graphics directory
e.g., C:\WINCALIS\GRAPHICS.
- In the directory in which the lesson is stored.

<p>Note: Graphic images stored in directories other than the WinCALIS Graphics Directory will only be available to scripts stored in that same directory, unless the full DOS path to the file is specified in the <GRAPHIC: filename.bmp> command. However, specifying a path is not recommended, to preserve portability during distribution of your exercises.</p>

Graphics in WinCALIS Author

You use the Graphic Dialog Box to incorporate graphics into lessons. Once you have saved your graphics in the WinCALIS Graphics directory or in your current directory (the directory your current lesson is saved in), the available files will be visible in the Graphic Dialog Box.

To Incorporate Graphics into Lessons:

- Go to the window (or dialog box) in which you will place the graphic.
- Choose Graphic under the Tools menu. A dialog box will appear listing the names of available graphics.
- Highlight the name of the graphic to place.
- Click on the OK button.

Note: Be Careful to test all graphics; large bitmaps may cover important text or extend beyond screen boundaries.

Notice that when you place graphics in WinCALIS Author, they are not visible. You will only see the CALIS command string. When a graphic has been inserted in a window, a status line will appear in the status bar of that window informing you what graphic will appear.

More Detailed Information on Graphics

Graphics are pictures displayed on the screen as part of WinCALIS exercises. They can be created with a graphics editor; copied from any of numerous electronic “clip art” collections; scanned in with an optical scanner from a printed picture; or be a combination of copying or scanning and further modification with a graphics editor. Each graphic is stored in a separate file on disk. WinCALIS Author does not display the graphic directly. Rather, when the user selects a graphic in Author, a command of the form <GRAPHIC: filename.bmp> is inserted in the desired Author window and saved as part of the WinCALIS script. The graphic itself is not stored as part of a CALIS script, but must accompany the script during distribution. Then when WinCALIS executes the script and encounters the <GRAPHIC> command, it retrieves the necessary graphics file from disk and displays it to the student in the specified window at the current cursor location.

What Graphics Can be Used in WinCALIS Exercises?

WinCALIS supports Windows standard Device Independent Bitmaps (DIB), which have the extension “.BMP.” “Device Independent” means that a graphic can be displayed regardless of the graphics card installed on the computer the student is using. Color graphics will usually be displayed on a monochrome monitor as shades of gray.

WinCALIS handles monochrome (black and white) and 16-color graphics up to approximately a full standard VGA screen (640 x 480 pixels) in size. WinCALIS also provides support for 256-color graphics, but the author should bear in mind that many students' displays will only handle 16 colors, and 256-color graphics are not as well-rendered in 16-colors as original 16-color graphics. Small-sized graphics are usually most effective. Assuming that WinCALIS screens will also contain text, small graphics are adequate for the great majority of applications. Considerations of screen design, disk space, and uncertain student workstation display speed all encourage the WinCALIS courseware author to make his graphics small and compact.

WinCALIS can display graphics in any window, including the Interactive (Question) Window, Presentation (Text) Window, Popup (Directions) Window, Diagnostic (Directed Feedback) Window, Lesson Description (Comment) Window, and Tutorial F[1-10] Windows. If the height of a graphic exceeds the size of the window in which it is displayed, a vertical scroll bar will appear and the graphic can be scrolled up and down, along with any text in the window. However, the need for frequent scrolling is not desirable, since WinCALIS does not always scroll an image 100% properly, particularly when the window is scrolled a line at a time with the up and down arrows on the scroll bar. Text and graphics can occupy the same position, so text can be superimposed successfully on a graphic of contrasting color, if the courseware author desires. (Text can also be created as part of the graphic itself.) On the other hand, since the student's answer line is centered in the Interactive (Question) Window and that line is fully rewritten with each new student keystroke, it will overwrite the part of

any graphic on the same line with the current foreground and background color, which looks unattractive. Hence, the author should try to place graphics in the Interactive (Question) Window either entirely above or entirely below where the student answer line is positioned.

Sources of Graphics

A fully-installed WinCALIS 2.x Author Workstation includes a small number of graphics in the default graphics directory--typically those needed in the sample scripts which come with the package. They have the names their creators originally gave them. Authors are free to copy, rename, and adapt any of these they desire for use in their own scripts. But these are primarily suggestive of what works well in WinCALIS and what the author can probably do as well or better on his own with a graphics editor.

Effective, attractive, and culturally sensitive graphics at a low cost may be obtained from resources that you might not immediately think of. For example, included in some installations of the WinCALIS program and on the Humanities Computing Laboratory Internet Home Page are hand-made illustrations by a local [student] artist. The pieces were created for inclusion in certain language exercises, but may be used by others. These graphics were drawn on paper with pen, and then scanned for conversion to electronic (*.BMP) format.



Figure 19: Sample Illustrations for Courseware Graphics

Computer graphics can be drawn using the drawing tools of a graphics editor. Numerous graphics editors are available, including many good shareware packages. The Windows graphics editor "Paintbrush," in the Program Manager Accessories group (or "Paint" in Windows 95/NT), is a good basic editor available to all Windows users. It stores files by default in Windows *.BMP format. No conversion of Paintbrush files is necessary for use in WinCALIS. If the Paintbrush is used mostly for creating WinCALIS graphics, it will be most convenient to set its working directory to the WinCALIS default Graphics subdirectory, under the main WinCALIS program directory. (Click once on the Paintbrush icon to highlight it, then click on the Program Manager File menu, click on the item Properties, then for Working Directory, type the path to the Graphics subdirectory, for example c:\wincalis\graphics.)

If a graphics editor (or scanning software) does not permit saving graphics to disk directly as Windows *.BMP files, they must be converted before WinCALIS can use them. One easy method of conversion is by copying a graphic to the Windows clipboard, then pasting it into Paintbrush, and there saving it as a monochrome or 16-color *.BMP file. Care should be taken to ensure that the entire bitmap graphic, not just the colored foreground image, does not exceed the maximum size WinCALIS can handle. A white background is part of the graphic! If Paintbrush or Paint is used to process the graphic, it is necessary to select the menu item "Image Attributes" in the "Options" menu, and specify the width and height of the image. If the image attributes are larger than the actual image, you can use the "pick tool" to put a box around just the image size you want, then save the graphic to a file on disk using the "Edit,

Copy to" feature, rather than saving from the File menu in the usual manner. Most other graphics packages have similar methods of controlling size of graphics. No matter how small the foreground image is, it is the settings for total size which will determine the size of the graphics file on disk and, in turn, whether or not WinCALIS will display it in the space you expect.

If a scanner is used to capture existing printed pictures, bear in mind that details which rely on delicate contrast in color will not be visible in the final monochrome or 16-color bitmap graphic. Choose pictures with broad contrasts of light and dark and simple lines. Black and white line drawings, even with great detail, make excellent WinCALIS graphics. If a picture photocopies well on an ordinary photocopier, it will probably also scan well and look good in WinCALIS. Graphics editors are good at manipulation of images, such as expanding and shrinking them, so size of the original is not important.

How to Insert Graphics in WinCALIS Author

WinCALIS Author does not directly display graphics. It tells WinCALIS to display them via the <GRAPHIC> command. When WinCALIS executes the <GRAPHIC> command, it gets a graphics file from disk and imports it into a WinCALIS exercise. For example, the Interactive (Question) Window could have the following text in it:

Look at this map and then tell how far it is from London to Bristol.
<GRAPHIC: ukmap.bmp>

The upper left-hand corner of the graphics file will appear at the point at which the <GRAPHIC> command is inserted. The graphic file filename extension .BMP may be omitted. The command <GRAPHIC: ukmap> is the same as the command <GRAPHIC: ukmap.bmp>.

A graphic can be inserted in ANY window, including the Answers/Feedback dialog box feedback window and Lesson Description Window. The <GRAPHIC> command can be simply typed in manually in the appropriate Author window at the point where it should display in the corresponding WinCALIS window. Or it can be inserted automatically by using the Insert Graphic dialog box. The Insert Graphic dialog box will insert the <GRAPHIC> command, as well as some placement commands and commented instructions for modifying them.

The Graphic Dialog Box

To open and display the Graphic dialog box, click on the Utilities pull-down menu on the main menu bar and select the menu item Graphic. Before opening the Graphic dialog box, the window into which the graphic is to be inserted should be made the active window (with highlighted title bar). In addition, the user should usually be careful that the caret, or cursor, (showing the insertion point) is at the point in relation to his text (before, after, or somewhere in the middle) where the graphic is to be displayed, starting from its upper left corner. (Since in the Interactive (Question) Window, the student answer line must also be taken into account, see below for further details on handling graphics in the Interactive (Question) Window.)

The Graphic dialog box contains a list box showing all available graphic files (with the extension .bmp) in the current subdirectory (where the *.WCA author file is saved) and in the common Graphics subdirectory. To select a graphic, type the graphic name in the edit box above the graphic list box, or simply click on the graphic's name in the list.

When a graphic is selected, it is not visible in Author itself, but an information message of the form G:filename.bmp is inserted in the title bar for the task number and window containing the graphic (except in the case of the Answers/Feedback dialog box feedback window). In addition, the actual string of CALIScript language commands which will be inserted in the WinCALIS *.CAL script is inserted in the text of the window at the current insertion point (caret location). The number values in these commands may be edited by the user to fine tune the position of a graphic in relation to the text in

the window. In fact, editing is more often than not required so as to achieve the desired result, so some understanding of these commands is desirable.

Command Strings Which Author Inserts For You

Author inserts the following strings of commands in various windows:

- Directions and Presentation (Text) Windows:
 <CUR:7><TAB:4><GRAPHIC:filename.bmp>
- Lesson Desc. and D windows: <CUR:2><CE><GRAPHIC:filename.bmp>
- Interactive (Question) Window:
 <CUR:7><TAB:4><GRAPHIC:filename.bmp><CUR:2>

The <CUR:n> command moves the cursor to row n in the current window. The <TAB:n> command moves the cursor over to the nth tab stop on that row. The <CE> command centers the graphic in the row. Thus the initial position for a graphic in the Presentation (Text) Window is beginning at row 7, tab stop 4.

Most authors will prefer to adjust the n values in order to place their graphic more accurately, or even remove the <CUR:n> and <TAB:n> or <CE> commands entirely and move the <GRAPHIC filename.bmp> command manually with hard returns, tabs, and spaces, or cutting and pasting to the desired cursor location.

Graphics and the Student Answer Line in the Interactive (Question) Window

The presence of a student answer line, in the case of the short answer type, and the need to generate automatically student answer choices in true/false and multiple choice answer types requires special handling of the Interactive (Question) Window. The initial value of the <CURSOR> command in the Interactive (Question) Window, <CUR:7>, displaying the graphic starting on row 7, is meant to allow space above it for many typical answer types, including short answer, true/false, and multiple choice of 4 or fewer choices. Furthermore, note that in the case of the Interactive (Question) Window, the string of CALIScript commands ends by returning the cursor back up to row 2 with the <CUR:2> command, where it is then ready for the student's answer line input on the next line, row 3 (or for Author's automatic generation of choice lines for true/false and multiple choice answer types).

This is acceptable if the question lines occupy no more than rows 1 and 2. But if the courseware author chooses to begin the question a few lines lower, or the question extends over several lines, the initial value of 2 will certainly need to be increased. Similarly, if a multiple choice includes more than 4 choices, the initial value of <CUR:7> will need to be increased, in order to display the graphic further down in the Interactive (Question) Window. Or alternatively, the initial value of <TAB:4> will need to be increased or decreased, along with a decrease in the initial <CUR:7> command, in order to display the graphic alongside the multiple choices to the right or left.

Sizing WinCALIS Windows Correctly for Graphics

WinCALIS does NOT size the height of a window automatically based on the size of a graphic displayed in it. WinCALIS sizes windows based on text (letters and words) in the window. Since the Interactive (Question) Window, the Popup (Directions) Window, and the tutorial “F” windows fill the entire screen by default, no matter how little text there is in them, there is usually no need to compensate for this in displaying graphics in these windows. But since the Presentation (Text) Window, the Lesson Description window, and the Directed Feedback “D” window are dynamically

sized to fit the text displayed in them, it is frequently necessary to force WinCALIS to generate a larger window to accommodate a graphic in these windows.

In the case of the Presentation (Text) Window, if there is little or no text in the window and the window is thus too small for the graphic, the user can insert the CALIS <SIZE> command in the corresponding Presentation (Text) Window in Author to increase the vertical size of the window manually. E.g.,

`<SIZE T 0,0,50,100>`

This command will force the Presentation (Text) Window to occupy the top half of the screen. (Conversely, a Popup (Directions) Window or F window can also be downsized with the <SIZE> command to just fit text or a graphic displayed in it.)

Alternatively, the user can force WinCALIS to size a window correctly by inserting in the corresponding Author window dummy text consisting of a series of hard returns (with ENTER key), following by a nearly invisible period (.) or a non-displaying character such as the “zero-width space” (inserted by clicking on the Format menu, the Insert Submenu, then selecting “Zero width Space”) on the left margin or spaced over to the right. WinCALIS usually sizes a window horizontally correctly for a graphic, so only the height of the graphic needs to be estimated, and dummy text inserted. Experience and a little trial and error can lead to accurate and quick estimation.

Note that in windows such as the Directed Feedback D window and Lesson Description Window, which are not affected by the <SIZE> command, only the latter method of inserting dummy text will work to force WinCALIS to increase the size of the window. Thus if a graphic is inserted in the Feedback window of the Answers/Feedback dialog box, the Enter key should be pressed enough times and dummy text inserted to allow for the height of the graphic in the WinCALIS Diagnostic Window.

The WinCALIS Common Graphics Subdirectory

A graphic used in a script should reside either in the current directory (where the *.WCL script itself is located) or in the common Graphics subdirectory. The Graphics subdirectory is where all graphics which are part of the full Author Workstation package are installed. It can be used to hold an unlimited number of graphics files. It is particularly useful when such files are shared by multiple scripts in different directories.

The path to the common Graphics subdirectory is defined in the WINCALIS.INI file in the main WinCALIS program directory. It can be redefined by the author for his Author Workstation or by the system administrator for a Student Workstation to whatever path is desired by carefully editing the line

Graphics= ...

in the WINCALIS.INI file with a plain text editor, such as the Windows Notepad or the DOS “EDIT” program.

When a script requires a *.BMP file, WinCALIS first checks the directory defined by the Graphics= line in WINCALIS.INI, then checks the “current” directory (the directory in which the *.WCL script file resides) or the path specified in the <GRAPHIC> command in the script.

See Also: Setting the Graphics Subdirectory

WinCALIS Multimedia Editor

The Multimedia Editor is a separate component of the WinCALIS Author Workstation. It is the device you use to manipulate audio, video, and animation media files for insertion into WinCALIS lessons. Typical sources for audio and video include digital audio *.WAV files, Video for Windows (*.AVI) digital video files, laserdisc video files, animation files, MIDI music files, and audio CDs. You may use the Multimedia Editor to record voice or other audio from a microphone or line input.

The Multimedia Editor acts as a link between your multimedia content and WinCALIS. You create clips from multimedia files using the Multimedia Editor, thereby making them accessible to WinCALIS and ready for incorporation into scripts. The only exception to this process involves *.WAV files, which can be played and recorded directly using WinCALIS, if you prefer. (See the section Special Audio Options for more information on *.WAV files.)

Start the Multimedia Editor

You can access the Multimedia Editor in one of two ways:

- Click on the WinCALIS Multimedia Editor icon which appears in the Pangaea group window in Program Manager OR
- Choose the Multimedia Editor under the Tools menu in WinCALIS Author.

The Multimedia Editor will scan your Windows installation for all installed MCI drivers. When it is done, you will note that under the menu bar is an area reserved for reviewing the audio or video source. Below it are various control buttons and boxes in which to type and edit clip names. At the very bottom, a status line gives information as to the function of each button as you pass the mouse arrow cursor over it.

Multimedia Editor Menus

On the menu bar at the left is the "File" menu. Use this menu to open and save *.MMD clip files. If you wish to add to or revise an existing *.MMD clip file, click on "File, Open," go to the subdirectory which contains the file, and click on or type the filename and press Enter to open the desired clip file. The Multimedia Editor will automatically attempt to open the first device specified in the clip file, which will be listed in the "Device Name" box. To add new clips or new media sources (devices) follow the procedure below.

Selecting a Device

To create a new *.MMD clip file, you will start by first designating the type of media, or device, you wish to review. On the menu bar to the right of the "File" menu you will see a "Device" menu. Open this menu and select the device type you wish to review. For the purposes of illustration, let us say you are going to view and mark up an existing Video for Windows digital video file, which was created using the software which comes with the Video for Windows package, the Video Blaster video card, or another of many available utilities for capturing and digitizing video. You would select the device "Video for Windows." This will open up an "Open" dialog box in which you have the opportunity to open a Video for Windows *.AVI file. Open the file.

Example: NEDBRAU.AVI and BEERCOM1.CAL

Let us say that the file you open is a digitized one-minute beer commercial, named "nedbrau.avi," for which you are constructing a series of scripts, starting with one entitled "beercom1.cal." After you have opened "nedbrau.avi," the video image should appear in a window in the upper part of the screen. The total length in frames (at 15 frames per second) is indicated on the right side of the screen, while the

current position is indicated on the left side of the screen. Use the tape recorder-style controls at the bottom of the control area to play, stop, rewind, and so on. Note that the slider in the horizontal control bar also indicates the current position in the media source.

Note also that the media source was opened by selecting a device from the "Device" menu, not the "File" menu. In the Multimedia Editor, the "File" menu is used for opening and saving *.MMD clip files, with the single exception that it is also used for saving a newly recorded digital audio *.WAV file, before saving a *.MMD file, before opening a new device or new *.MMD file, or before exiting the Multimedia Editor. Each time you wish to open a new media source of any kind, you should go to the "Device" menu and select the desired device type.

Note: If you have already worked on a multimedia file for a particular WinCALIS script, be sure to open *that* one if you plan to add more multimedia to that script.

Note: Your *.wca script file and your multimedia *.mmd file must have the same filename.

Editing in the Multimedia Editor

At any time after you have created an *.MMD file, you may edit it by deleting clips, defining new ones using the same file or by adding clips using different files or devices. Once you have created a file, the procedure to access it is the same as for opening any other file.

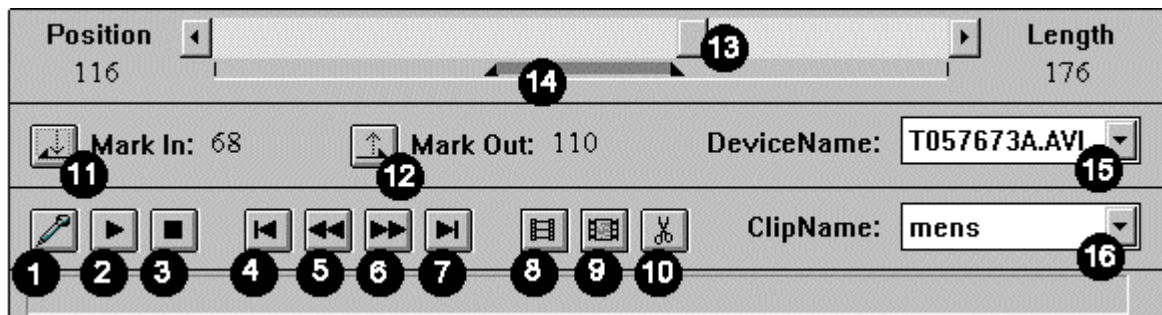


Figure 20: Multimedia Editor Controls

Multimedia Editor Controls:

1. Record
2. Play Device
3. Stop Device
4. Rewind (to beginning of device)
5. Rewind (1/16 of device length)
6. Fast Forward (1/16 of device length)
7. Fast Forward (to end of device)
8. Name Clip
9. Edit Clip
10. Delete Clip
11. Mark In Button
12. Mark Out Button
13. Slider
14. Clip Indicator
15. Device Name (e.g.: *.avi video or *.wav audio file[s].
*.mmd files may contain multiple devices)
16. Clip Name (each device may contain multiple clips)

To open an *.MMD file:

- In the Multimedia Editor, choose Open under the file menu. Change to the directory which contains your lesson. A dialog box will appear listing the names of all the *.MMD files in that directory.
- Select the file you want to open by highlighting it or by typing the filename in the File Name dialog box.
- Click on the OK button.

The Multimedia Editor will automatically display the first clip you marked in that file. Remember, only the clips that you defined using that particular file will be displayed in the ClipName box. Once you have located the file that you want to make changes to, you can proceed as usual. You mark clips in the same manner as when you are first working on an *.MMD file. You could also create new clips from different files by choosing the device from the Device menu and continuing as described in the section Creating *.MMD files.

**To create new clips in the Multimedia Editor
(of audio or video format) :**

- Select the device the media file is stored in from the Device menu which appears on the menu bar at the top of the Multimedia Editor window. A dialog box will appear allowing you to open one of the files listed.
- Open the file that you will create clips from. If the file is an animation or video file, the video image should appear in the upper part of the screen. If the file is an audio file, a window will appear in the middle of the screen with a graphic image of the sound.

Once you have opened the file, you are ready to begin marking clips. The tape recorder style controls at the bottom of the screen are used to manipulate (play, stop, fast-forward etc.) the multimedia file. The slider in the horizontal control bar indicates the current position in the media source. The Mark In and Mark Out markers are located at the leftmost side of the horizontal scroll bar before you begin.

- Move the slider to the point at which you will begin your clip. This can be done by playing or fast-forwarding to the desired location, or by dragging the slider to it.
- Mark the beginning of the clip by clicking on the Mark In button found on the left-hand side of the window.
- Position the slider at the location you will end the clip by playing, fast-forwarding or dragging the slider.
- Click on the Mark Out button found to the right of the Mark In button in order to designate the end of your first clip.

At times you may want to create a clip which is actually the entire file. In order to mark the entire media file as one clip:

- Position the slider at the beginning of the recording.
- Click on the Mark In button.
- Drag the slider to the end of the video, audio or animation recording.

- Click on the Mark Out button.

Once you have marked your first clip, the Multimedia Editor will ask you to name it. The clip name itself will not carry an extension, it is just one part of the *.MMD file which comprises the entire media file and all the defined clips. The clip name can be up to nineteen characters long with no spaces.



To name a clip:

- Click on the Clip Name button, which appears as a strip of film in the lower right-hand side of the window. A Clip Name dialog box will appear.
- Name the clip any name up to nineteen characters long without spaces and with no extension.
- Click on the Add button which appears in the dialog box. The name of the clip you just entered will appear highlighted in the Clip Name selection box.

Important: When making changes remember always to check whether the selection which you are going to modify has first been highlighted. Otherwise you might accidentally change some other selection.

To play a clip:

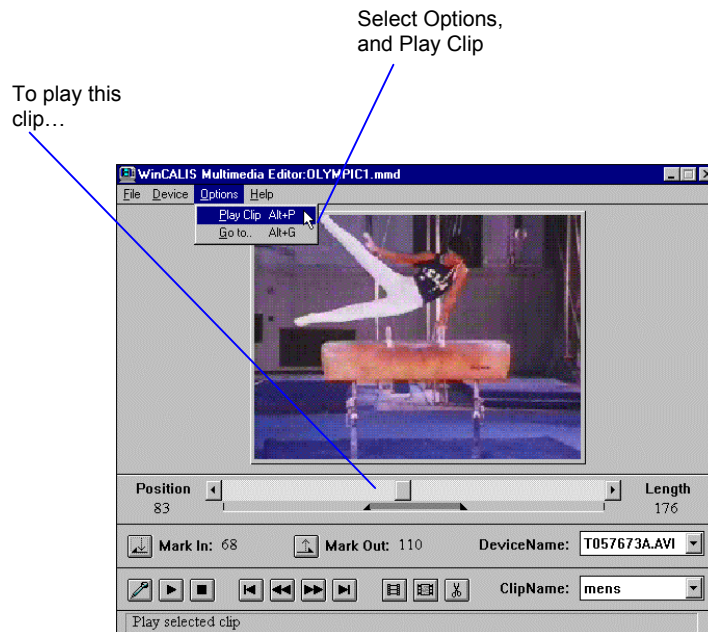


Figure 21: To play a clip (section of a multimedia file)

- To play the selected clip or device section (indicated by the gray bar), choose the Options menu item, and Play Clip (Alt+P).

Note: If in the same *.MMD (for a single script) you switch to a different device (such as a different *.WAV file), you will no longer see the clip names you already defined from the previous device. However, those clips still exist. Therefore, do not save new clips with the same name as any other clip in that *.MMD file. (WinCALIS Author will display all audio clips from all audio devices in the *.MMD file together and all video clips from all video devices together.)

You may continue marking clips using any number of files or devices. Each time you want to create a new clip with a file different from the one you are currently working on, go to the Device menu and choose the device the file is saved under. You follow this process whether you are creating another clip from a different file under the same device, or creating a clip from a different device altogether.

Saving clips

Once you are finished creating all the clips you will place into a lesson, you must save them in an *.MMD file with the same name as the lesson in which you will be placing them. If you do not save the file with the same name, you will not be able to incorporate the clips into your lesson.

To save clips:

- Once you have created all the clips you want, click on Save in the File menu. A Save dialog box will appear.
- Enter the name of the lesson you will incorporate the clips into with the *.MMD extension. For instance, if your lesson was entitled FIRSTLES.WCA you would name your *.MMD file FIRSTLES.MMD.
- Click on the OK button.



To redo a clip:

- Set the NEW Mark In and Mark Out at the desired. Click on the redo clip button, and then click on the clipname (that you wish to redo).



To delete a clip:

- Set the NEW Mark In and Mark Out at the desired. Click on the redo clip button, and then click on the clipname (that you wish to redo).

To Edit an Existing *.MMD file

At any time after you have created an *.MMD file, you may edit it by deleting clips, defining new ones using the same file, or by adding clips using different files or devices. Once you have created a file, the procedure to access it is the same as for opening any other file.

To open an *.MMD file:

- In the Multimedia Editor, choose Open under the File menu. A dialog box will appear listing the name of all the *.MMD files. If necessary, change to the directory which contains your script and its accompanying *.MMD. file
- Select the file you want to open by highlighting it or by typing the filename in the File Name box.
- Click on the OK button.

The Multimedia Editor will automatically display the first clip you marked in that file. Remember, only the clips that you defined using that particular file will be displayed in the Clip Name box. In order to access any other files, you must know the type of device (*.AVI, *.WAV, etc.). If you do not remember the filename, you can find it in WinCALIS Author. In Author under the Tools menu, you can choose the device that the file is saved under. A dialog box will appear listing the clip names under the file it originated from.

Once you have located the file that you want to make changes to, you can proceed as usual. You mark clips in the same manner as when you are first working on an *.MMD file. You could also create new clips from different files by choosing the device from the Device menu and continuing as described in the section "Creating *.MMD files."

Saving the *.MMD Clip File

When you are finished defining clips, click on "File, Save," to save the multimedia clip file (with *.MMD extension). Be sure to save the clip file in the same directory as the one which holds your script. For example, if your script is named "beercom1.cal" and is saved in the directory "c:\WinCALIS\scripts\german3," then change to that same directory and save your clip file as "beercom1.mmd." If your media source is a newly recorded waveform audio (*.WAV) file, you will be prompted to save it first, with a *.WAV file extension. Then you will save the multimedia clip file, using the same 1-8 letter filename as the name of the script which the audio is meant to accompany. If you save the *.MMD file in a directory other than the one which holds your script, you will need to use the Windows File Manager to move the clip file to the directory with your script. To exit the program and return to the Program Manager or WinCALIS Author, select "Exit" from the "File" menu.

This process produced a small text multimedia clip file (*.MMD file), in addition to the usually much larger video or audio media file itself or the external simple media device, with information about the starting and stopping points of your audio clips. It has the same name as your WinCALIS lesson, but with a *.MMD extension. You may at this point go back to the WinCALIS Author program, open your script and incorporate the video or audio you have just created into your script.

The Structure of the *.MMD Clip File

Remember that all the video and audio media clips for a given WinCALIS script are defined in a separate plain-text clip file with the extension *.MMD. It is important to be aware of the contents of the clip file, especially in connection with transferring your script to a student workstation or preparing your multimedia scripts for general distribution. The clip file will probably need to be edited to ensure portability.

The *.MMD clip file produced by the Multimedia Editor is a tiny ASCII file with a simple, precise syntax. It must have the same 1-8 letter filename as the WinCALIS script itself, and be contained in the same directory as the script file. Although the Multimedia Editor automatically handles for you the creation and editing of the *.MMD clip file, it may be useful for you on occasion to edit or view an *.MMD file directly, or even to create a new one from an existing one or from scratch.

One case in which you would need to edit an *.MMD file is when you wish to move the multimedia source file from one drive or directory to another, and you need to specify the new path to the file in the *.MMD file--perhaps a large digital video file formerly resided on the C: drive on a hard disk, but was subsequently transferred to a CD-ROM, played in a CD-ROM drive with the drive letter G:. No path is necessary if the multimedia file is in the same directory as the script (the "current directory") or in the directory specified in the Multimedia= line in the WINCALIS.INI file. Thus you may wish to edit the DOS path specified in the device line, leaving just the DOS filename.

Editing the *.MMD File

To edit an *.MMD file, use the Windows Notepad or other plain text editor, such as the DOS Edit program. You may also edit the *.MMD file by opening it as a Document File in the WinCALIS Author Scratchpad, but since it is an ASCII-coded file (i.e., restricted to the lower 128 keyboard characters of the ANSI /Windows character standard, be sure to select the file type "ANSI/WinCALIS 1.1" in the "List Files of Type:" list box when opening the file, and likewise, be sure the same type is selected in the "Save File As Type:" list box when you save the file.

There can be an unlimited number of audio or video devices and clips defined in a single *.MMD file, and each device will have a section with the following structure: on the first line there is a device name specified in square brackets. If the device is a digital media file on disk (a so-called "compound device"), it is simply listed with its full DOS pathname:

```
[F:\PROGS\WINCALIS\MLTMEDIA\JOHNSON1.AVI]
```

Here are examples of device names for simple (hardware) devices:

```
[VIDEODISC]  
[CDAUDIO]
```

On subsequent lines there is on a single line a clip name, followed by an equals sign, followed by the frame or time in milliseconds from the beginning of the device where the clip begins, followed by a comma, followed by the frame or time in milliseconds from the beginning of the device where the clip ends. Spaces are ignored. E.g.:

```
[POLISHWD.WAV]  
duzy=000, 2000  
bialy=2000, 4000  
wlaczyc=4000, 6000  
goraco=6000, 8000  
poludnie=8000, 10000  
kobieta=10000, 12000  
otworzyc=12000, 14000  
ozenic=14000, 16000  
bez=16000, 17000  
ugotowan=17000, 19000  
entire=000,19000
```

(Windows power users will note that the syntax of the *.MMD file follows the standard syntax of a Windows *.INI file.)

Inserting Multimedia in WinCALIS Lessons

Once you have created and saved your clips under a file with the extension .MMD, you may insert them into your lesson.

Inserting Audio

To insert an audio clip:

- Go to the task and window into which you want to insert the clip. For instance, if you want an audio clip to play as part of the question in Task 2, you would go to the Interactive (Question) Window for Task 2.
- Click on Audio Clip... which appears in the Tools menu. A large dialog box will appear.
- Type or select the clip name you want to insert.
- Select the options you want to include. For example, you may choose to play both channels of a stereo audio clip, the left channel only, or the right channel only.
- Click on the OK button.

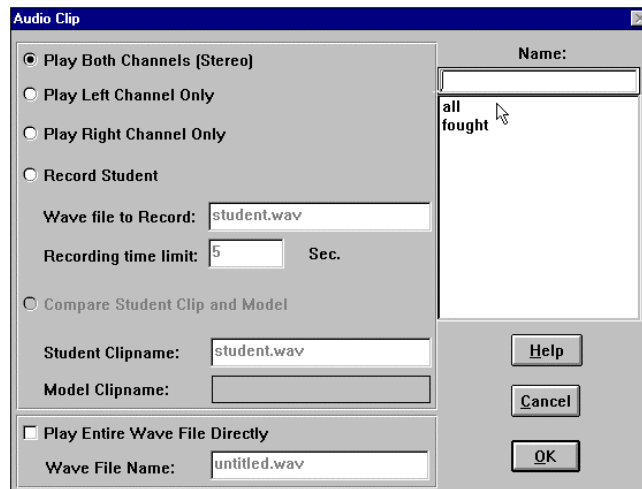


Figure 22: Insert Audio Clip Dialog Box

When you select an audio clip, WinCALIS Author will automatically insert into the currently selected task and window the commands needed for WinCALIS to play that clip, when you save your *.WCA author database file and generate the *.CAL script file.

You will not actually see the audio commands in your Author window. However, if the Script Generation Option “Info line in Task Windows” (Script menu) is enabled (the default), you will see a reminder that an audio clip has been inserted (of the form “A:clipname”) in the title bar of the window for the task which contains the clip.

Note: When you type in the name of a *.WAV file to play directly in the lesson, WinCALIS does not check to see if that file actually exists or the accuracy of the filename.

See Also: Audio Options

Inserting Video or Animation

To insert video or animation:

- Go to the task and window into which you want to insert the clip. For instance, if you want a video clip to appear in the directions before Task 3, you would go to the Popup (Directions) Window for Task 3.
- Click on the Tools menu.

- Select the menu item Video Clip. A dialog box will appear which contains a list of video clips which have been defined and various display options. Highlight or type the name of the clip you want to insert and select your options.
- Click on the OK button.

When you select a video clip, WinCALIS Author will automatically insert into the currently selected task and window the commands needed for WinCALIS to play that clip, when you save your *.WCA author database file and generate the *.CAL script file.

You will not actually see the video commands in your Author window. However, if the Script Generation Option "Info line in Task Windows" (Script menu) is enabled (the default), you will see a reminder that a video or animation clip has been inserted (of the form "V:clipname") in the title bar of the window for the task which contains the clip.

See Also: Video and Animation Options

Note: If you try to insert audio or video into a new WinCALIS script that has not yet been saved, the Author will be unable to find any audio or video clips since it looks for an *.MMD file with the same name as the script. You must first save your script (with the same name as the *.MMD file containing clips for that lesson). Be sure that there is also an *.MMD file in the same directory as your script. However, in the case of typing the filenames for digital audio *.WAV files in the Audio Clip dialog box, Author merely accepts whatever filename you give, without verifying whether or not such a file actually exists. Caveat auctor!

A Detailed Example of Inserting Video Clips

Using the Audio and Video Clip Dialog Boxes to Insert Multimedia In Your Exercise

Let us assume that you have already begun a script named "beercom1.cal," which is designed around viewing and understanding a beer commercial for "Nedbrau" beer. You have on your disk the one-minute beer commercial in the form of a digital video file, named "nedbrau.avi." You have already used the WinCALIS Multimedia Editor to view the commercial and mark a series of "clips" you want the student to see and hear at various points. One of the clips, "entire" includes the whole commercial, while other clips, "he_quaffs," "she_smiles," "frothy," "price_is_right," and so on mark the individual segments the student may need to see. The files "nedbrau.avi" and "beercom1.mmd" are saved in the same subdirectory which contains your script "beercom1.cal" and Author file "beercom1.wca."

Now you are ready to insert multimedia in your script. Preferably you have made notes specifying what the clips are, and where they should be inserted in your script. Now you wish to go back to your script and actually insert the clips at the suitable points, indicating as you do so the various options for viewing them, such as their exact location on the screen, their size, whether to crop the image so as to show just a portion (perhaps cutting off subtitles or focusing attention on a particular cultural aspect), whether to play just the video portion or just the audio portion, and so on.

In WinCALIS Author, open the Author file "beercom1.wca" for your beer commercial script "beercom1.cal," if it is not already open. Go to the window in which you wish to insert a video clip for the student to view. Let us say you wish the student to view the entire commercial without interruption at the outset. You could do this by putting the clip in the Directions window, the Question window, or Text window for the first task. Let us put it in the Directions window. Go to the Directions window. Your written directions to the student will probably invite him to watch the whole commercial, before proceeding to do the other tasks you have in store for him. They might also remind the student that he can view the current video clip at any time by pressing the F12 function key or clicking on the item "Play Video" in the "Options" menu. Similarly the student can listen to the current audio clip at any time by pressing the F11 function key or clicking on the item "Play Audio" in the "Options" menu. If you turn over full control to the student for playing the video or audio, via the MCI Control Bar, your

directions might also give the student some road signs for recovering from getting lost in a lengthy clip from a laserdisc or videotape.

Inserting a Video Clip

Now click on the "Utilities" menu on the menu bar and select the item "Video Clip." WinCALIS Author will look in the current directory (the directory in which your script is to be found) for the file "beercom1.mmd." If it finds it, it will scan it for all video-related devices and display their respective clip names. In a similar manner, if you are inserting an audio clip, you would select the item "Audio Clip" from the "Utilities" menu, and Author will scan the same file "beercom1.mmd" for all audio-related devices and display their respective clip names.

If your script is a new one which has not yet been saved to disk and is thus nameless, Author will warn you to first save it to disk. If Author cannot find a clip file with the same name as your script in the current directory, it will warn you to verify the location of the clip file. You may need to use the Windows File Manager or other utility to locate your clip file and move it to the directory containing your script. If you also move a media file itself to the directory containing your script from another location where it was reviewed and marked up using the Multimedia Editor, you will need to edit the DOS path specified for that device in the *.MMD clip file. Open the clip file in the Windows Notepad or other plain-text editor and carefully delete the entire path except for the filename itself. E.g., you would leave just "[BEERCOM1.MMD]" as the device name. See the preceding section "The Structure of a *.MMD Clip File" for more information on *.MMD clip files and their binary distribution versions, *.WCM files.

In the Video Clip dialog box you should see the clip names you have defined displayed in the list box on the right:

```
entire
frothy
he_quaffs
price_is_right
she_smiles
.
.
```

Click on the clip you wish the student to see, in this case the clip named "entire," which marked the whole one-minute commercial. You will see the selected clip name "entire" appear in the box at the top. The default values for various viewing options are indicated at the left side of the box. You may change the default values by clicking on the radio button you wish to select or on one of the buttons which opens a secondary dialog box for you to indicate your further preferences. This time click on OK or press Enter to accept all the default values. You will return to the window from which you entered the Video Clip dialog box, in this case the Directions Window. You should see the message "V:entire" appear in the title bar of the Directions window for Task 1. The default values for a video clip are as follows:

- stretched or shrunk to a quarter screen in size
- located in the center of the screen;
- to be played and then hidden;
- with full soundtrack;
- played from the video source connected to the port COM1 (in the case of external video devices connected to COM ports);
- showing 100% of the video image;
- displaying the "MCI Control Bar" in WinCALIS giving the student control over the video.

Let us say that the student responds incorrectly to your first question, which asked the student if "Nedbrau" beer appeared to be a light or dark beer. In this case you wish to give the student some feedback in the Directed Feedback "D" window, showing the student a brief close-up of this very dark beer being poured into the glass, along with some text feedback helping him with the concepts "light" and "dark" in the target language. (Although video is inserted in a "window" it is always shown in relation to the WinCALIS screen as a whole.) You previously gave this scene the clip name "frothy." Now go to the Question window for Task 1 and in the "Task" menu, select the answer type you have designed for this task. Let us say it is a "Short Answer" question. Open the "Short Answer" Answers/Feedback dialog box. Go to the Incorrect/Remain Here anticipated response (the word meaning "light"), in reaction to which you wish to show the student the clip "frothy." Click in the Feedback box to make it active, type your text feedback, then click on the "Video Clip" item in the "Utilities" menu, and this time select the clip "frothy" and click on OK to insert the video clip "frothy" as visual feedback, if the student gives this Incorrect/Remain Here answer. Note that in this case there is no message in the title bar or other cue from Author that the desired video commands will be generated.

Inserting an Audio Clip

In the case of inserting an audio clip using the Audio Clip dialog box, the procedure is the same. You click on one of the listed audio clips to select it. Then you accept the default values for listening options or select different ones, then click on OK or press Enter. When you have inserted an audio clip, you will see the message "A:clipname" appear in the title bar of the window in which the audio clip is inserted. The options for audio clips are much simpler than for video clips. The default value is simply to play both channels. However, in the case of digital audio *.WAV files, there is an additional option to play a *.WAV file directly, without having previously created a clip file or defined clips for it. This option can be freely used by the courseware author who does not wish to define clips, but it is the responsibility of the courseware author to ensure that the named *.WAV file actually exists at the time WinCALIS executes the script.

WinCALIS Default Settings for Multimedia

When WinCALIS executes a video or audio "play" command, it will use the default settings for any play option (size, location, etc.) left unspecified. But once a command has been executed changing the default setting to something different, that new setting will remain in effect until changed. For example, once "full-screen" video is called for, WinCALIS will continue to play all video full-screen until instructed to do otherwise. In practice, WinCALIS Author generates many redundant commands in a WinCALIS script through the "Video Clip" and "Audio Clip" dialog boxes. The courseware author who enters video or audio commands directly in the script, however, can avoid such redundancy and save himself much labor by recalling this principle.

Audio and Video Options

When inserting audio, video or animation files, you may choose a variety of options that affect various aspects of the clip.

Audio Options

Your options for audio clips are fairly limited. Because audio clips deal only with sound, those are the only options you can control. You may choose to play both channels of a stereo audio clip, the right channel only, or the left channel only.

Special Audio Option: Record Student

WinCALIS provides a feature which allows authors to invite students to record a sound file through a microphone. To do this, activate the window at the point that the option should become available (e.g.

the Interactive [Question] Window of Task 4). Select Tools, and Audio Clip; you will see the Audio Clip dialog box. Choose the “Record Student” button, which will then activate the other associated options. Use the default file name and record time, or overwrite them with your own selections.

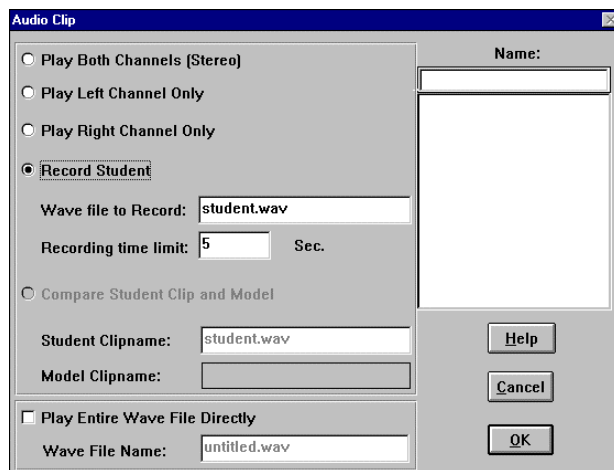


Figure 23: Choosing "Record Student" in the Audio Clip dialog box

Playback of a student-recorded audio file

The usual way to incorporate multimedia into WinCALIS lessons is through defining clips in the Multimedia Editor. There is one exception to this rule in the case of *.WAV files. The option “Play Entire Wave File Directly” is used in conjunction with the Record Student option, allowing you to record the student in a *.WAV file as part of the lesson, and play back the file for him at a later point.

The “Record Student” feature saves temporary audio files to one of two places. It will first try to place it in the current subdirectory (that is: from where the script is being run). If this is unavailable, as in the case of a script being run from a non-recordable media such as a CD-ROM, then WinCALIS will place the temporary audio file in the \MLTMEDIA subdirectory of the installation directory.

Note: When you type in the name of a *.WAV file to play directly in the lesson, WinCALIS does not check to see if that file actually exists or the accuracy of the filename.

Note: Playing an Entire Wave File Directly limits options in the final script presentation (namely: no MCI Control, and no Play [Last] Audio F11 function key), and should only be used in the case of playing back a recorded student. For pre-prepared audio files, insert clips as defined in the Multimedia Editor.

Video and Animation Options

Your options for controlling the appearance of video and animation clips are quite extensive. These options are all listed on the left-hand side of the Video Clip dialog box, which appears when you choose Video Clip in WinCALIS Author. These options are explained below.

Size

The Size box allows you to specify the size of the visual clip on the student's screen. You may specify Full Screen, Quarter Screen, Eighth Screen, or Sixteenth Screen. For maximum flexibility in both size and location, you may also press the **Customize Area** button, found at the bottom of the dialog box. This option opens another dialog box which allows you to specify more precisely the size, location, and even the shape of the image on the student's screen. Specifying *shape*, however is more an obligation with this feature than a bonus--it is your responsibility to keep the horizontal and vertical dimensions in proportion, so that the image is not unduly distorted.

Note: If you select the Full Screen option, the Location buttons will be dimmed. This occurs simply because if you want the image to appear on the entire screen, you cannot specify a location.

Location

You use the Location option to specify the area on the screen in which you would like the clip to appear. You may specify Top Left, Top Center, Top Right, Center, Lower Left, Lower Center, or Lower Right. If you click on the **Specify Location** button, it will open a Specify Location dialog box, in which you will be able to determine the location of the clip on-screen with more accuracy, while using the size selected in the Size box. For the most flexibility, the **Customize Area** option enables you to define the size, shape, and location (see Size above).

Play

The Play box provides you with the option of playing and hiding the clip, playing and freezing it, or playing Audio Only. The Play/Hide option causes the image to disappear whenever the clip is stopped (by the student, if he or she has control) or when it has reached completion. The Play/Freeze option will freeze that image when it is stopped by the student, and also when it has reached the final frame.

Audio

The Audio option allows you to specify which audio track, if there is more than one, to play along with the video clip, or to play it without sound.

Video

You may specify to which port the desired external device is connected, if there is more than one.

Zoom Image

This option enables you to crop your image as you would a photograph. If you click on the **Zoom Image** button, it will open a Zoom Image dialog box, in which you will be able to specify the portion of the image to show. You could, for instance, choose to focus on one aspect of the image in your clip using this feature. Or you could mask out subtitles by cropping the bottom of the image.

MCI Control / Media Controls Bar



Figure 24: MCI Control Bar / Media Controls

When enabled, this option allows the student to manipulate the clip. He or she can play, pause, stop, fast-forward and rewind the image when this toolbar is showing. The student enables this option in WinCALIS under the Options menu.

The default values for a video or an animation clip are as follows:

- Size: Quarter Screen
- Location: Center
- Play: Play/Hide
- Audio: Both Tracks
- Video: COM1 (when external video devices are connected to COM ports)
- Crop Image: 100% of the image

- Student Controls: MCI Control bar displayed, so as to give student control of the image.

Note: Until the “Student Controls” option is enabled in the Video Clip dialog, if you wish to suppress the MCI Control Bar WinCALIS, you need to enter the CALIS command manually in your window. Just insert the command <MCICTRL: HIDE> after the first occurrence of an audio or video-related command in your script.

Summary of Audio Commands

Audio Options and their Corresponding CALIS Commands

Option..... Command Written to WinCALIS Script

Play Options

Play clip <AUDIO: clipname>
 Pause Audio <AUDIO: PAUSE>
 Resume Audio <AUDIO: RESUME>

Other Options

Play Both Channels (Stereo) <AUDIO: STEREO>
 Play Left Channel <AUDIO: LEFT>
 Play Right Channel <AUDIO: RIGHT>

Wave File Options

Play Digital Audio Wave File <AUDIO: PLAY filename.wav>
 Record Student <AUDIO: RECORD filename.wav milliseconds>
 Delete Wave File <AUDIO: DELETE filename.wav>

Summary of Video Commands

Video Options and their Corresponding CALIS Commands

Option..... Command Written to WinCALIS Script

Play Options

Play/Hide <VIDEO: PLAYHIDE clipname>
 Play/Freeze <VIDEO: PLAY clipname>
 Play Audio Only <VIDEO: PLAYSND clipname>
 Pause Video <VIDEO: PAUSE>
 Resume Video <VIDEO: RESUME>
 Show Video <VIDEO: SHOW>
 Hide Video <VIDEO: HIDE>
 Cue video to starting frame <VIDEO: CUE>

Size Options

Full Screen <VIDEO: SIZE FULL>

Quarter Screen<VIDEO: SIZE QUARTER>
 Eighth Screen.....<VIDEO: SIZE EIGHTH>
 Sixteenth Screen.....<VIDEO: SIZE SIXTEENTH>
 Crop Image<VIDEO: FRAME 0,0,100,100>

Location Options

Top Left<VIDEO: LOC TL>
 Top Center<VIDEO: LOC TC>
 Top Right<VIDEO: LOC TR>
 Center.....<VIDEO: LOC CE>
 Lower Left<VIDEO: LOC LL>
 Lower Center<VIDEO: LOC LC>
 Lower Right<VIDEO: LOC LR>
 Specify Location<VIDEO: LOC 0,0>
 Locate video at mark.....<VIDEO: LOC mark>
 Customize Area.....<VIDEO: SIZE AREA 0,0,100,100>

Video Soundtrack Options

Both Tracks.....<VIDEO: AUDIO STEREO>
 Track 1<VIDEO: AUDIO LINE1>
 Track 2<VIDEO: AUDIO LINE2>
 No Audio.....<VIDEO: AUDIO NONE>

Video Source Options

Source 1<VIDEO: LINE1> (<VIDEO: LASER>)
 Source 2<VIDEO: LINE2> (<VIDEO: VCR>)

Other Options

Student Controls.....<MCICTRL: SHOW|HIDE>

Distributing Multimedia Exercises

When the Author Workstation version of WinCALIS converts a *.CAL script file into a *.WCL file, it also converts the *.MMD file into its binary-encoded equivalent *.WCM file (if one does not already exist). When scripts employing multimedia are distributed for use on WinCALIS student workstations, it is essential to include the *.WCM clip file along with the *.WCL script file, and any other subsidiary files, including, of course, the digital audio or video files, CDs, laserdiscs etc., which the clip file references. The Student Workstation version of WinCALIS is capable of reading only *.WCL files, *.WCM files, *.WCI file (binary versions of tutorial files), *.GLU files and other such already-converted files. Since it is irritating to receive and try to work an exercise which is missing one or more required peripheral files, good authoring practice encourages keeping track of them by listing all the separate files, including graphics and multimedia, required for a given script in the *.CAL file itself (such as in a header comment in the Pass Through Window), or better (since the *.CAL file is not normally distributed to end-users and the *.WCL file is not directly viewable) in a separate ASCII text file. Some authors use the extension *.SUB for such a list. E.g., the script "infinito.wcl" would travel along with a parallel file "infinito.sub" listing all the subsidiary files required:

```
infinito.wcm
infinito.wav
arbol.bmp
```

audifono.bmp
bodega.bmp
cabana.bmp
.
.
vocab1.wci
engtrans.wci
etc.

Chapter 7:

Special Authoring and Editing Functions

Browsing and Inserting Text From Unicode Tables

Unicode Character Selection Dialog Box

The Unicode Character Selection dialog box enables you to browse through the Unicode tables for your general information. You may also insert a character in your text by going to its location in the table, either by scrolling through Unicode character blocks or by typing all or some of its hexadecimal Unicode character code in the edit box in the lower right corner.

Hint: Clicking above or below the slider (or “scroll box”) on the vertical scroll bar will scroll up or down a full “page” of 256 characters. To scroll up or down a half-page, so as to see all the characters on a page, press the up or down scroll arrows. If you move the focus to the window by clicking anywhere in the window, you may also move about with the cursor control keys. Pressing PageUp or PageDown will scroll up or down a full page, while the arrow keys will move about a row or column at a time.

The shortcut key to open the dialog box in WinCALIS Author is Ctrl+I.

Here is a list of Unicode character “blocks,” with their starting and ending hexadecimal Unicodes:

START	END	BLOCK NAME
0020	007E	BASIC LATIN
00A0	00FF	LATIN-1 SUPPLEMENT
0100	017F	LATIN EXTENDED-A
0180	024F	LATIN EXTENDED-B
0250	02AF	IPA EXTENSIONS
02B0	02FF	SPACING MODIFIER LETTERS
0300	036F	COMBINING DIACRITICAL MARKS
0370	03CF	BASIC GREEK
03D0	03FF	GREEK SYMBOLS AND COPTIC
0400	04FF	CYRILLIC
0530	058F	ARMENIAN
0590	05CF	HEBREW EXTENDED-A
05D0	05EA	BASIC HEBREW
05EB	05FF	HEBREW EXTENDED-B
0600	0652	BASIC ARABIC
0653	06FF	ARABIC EXTENDED
0900	097F	DEVANAGARI
0980	09FF	BENGALI
0A00	0A7F	GURMUKHI
0A80	0AFF	GUJARATI
0B00	0B7F	ORIYA
0B80	0BFF	TAMIL
0C00	0C7F	TELUGU
0C80	0CFF	KANNADA
0D00	0D7F	MALAYALAM
0E00	0E7F	THAI
0E80	0EFF	LAO

10A0	10CF	GEORGIAN EXTENDED
10D0	10FF	BASIC GEORGIAN
1100	11FF	KOREAN HANGUL JAMO
1E00	1EFF	LATIN EXTENDED ADDITIONAL
1F00	1FFF	GREEK EXTENDED
2000	206F	GENERAL PUNCTUATION
2070	209F	SUPERSCRIPTS AND SUBSCRIPTS
20A0	20CF	CURRENCY SYMBOLS
20D0	20FF	COMBINING DIACRITICAL MARKS FOR SYMBOLS
2100	214F	LETTERLIKE SYMBOLS
2150	218F	NUMBER FORMS
2190	21FF	ARROWS
2200	22FF	MATHEMATICAL OPERATORS
2300	23FF	MISCELLANEOUS TECHNICAL
2400	243F	CONTROL PICTURES
2440	245F	OPTICAL CHARACTER RECOGNITION
2460	24FF	ENCLOSED ALPHANUMERICS
2500	257F	BOX DRAWING
2580	259F	BLOCK ELEMENTS
25A0	25FF	GEOMETRIC SHAPES
2600	26FF	MISCELLANEOUS SYMBOLS
2700	27BF	DINGBATS
3000	303F	CJK* SYMBOLS AND PUNCTUATION
3040	309F	HIRAGANA
30A0	30FF	KATAKANA
3100	312F	BOPOMOFO
3130	318F	HANGUL COMPATIBILITY JAMO
3190	319F	CJK* MISCELLANEOUS
3200	32FF	ENCLOSED CJK LETTERS AND MONTHS
3300	33FF	CJK* COMPATIBILITY
3400	3D2D	KOREAN HANGUL (Unicode 1.1)
3D2E	44B7	HANGUL SUPPLEMENTARY-A (Unicode 1.1)
44B8	4DFF	HANGUL SUPPLEMENTARY-B (Unicode 1.1)
4E00	9FFF	CJK* UNIFIED IDEOGRAPHS
AC00	D7A3	KOREAN HANGUL (Unicode 2.0)
E000	F8FF	PRIVATE USE AREA
F900	FAFF	CJK COMPATIBILITY IDEOGRAPHS
FB00	FB4F	ALPHABETIC PRESENTATION FORMS
FB50	FDFD	ARABIC PRESENTATION FORMS-A
FE20	FE2F	COMBINING HALF MARKS
FE30	FE4F	CJK* COMPATIBILITY FORMS
FE50	FE6F	SMALL FORM VARIANTS
FE70	FEFE	ARABIC PRESENTATION FORMS-B
FF00	FFEF	HALFWIDTH AND FULLWIDTH FORMS
FFF0	FFFD	SPECIALS

* “CJK” stands for Chinese-Japanese-Korean. For further Chinese-Japanese-Korean help, see also Kangxi Radical Chart (Chinese-Japanese-Korean)

See also: WinCALIS Author and Unicode

Pasting Text With Code Conversion

Use the Paste as... feature under the Edit menu to paste non-Unicode text from the clipboard, which was copied there in another application, such as a World Wide Web browser, into WinCALIS Author

at the insertion point (where the blinking “caret” is located), converting the codes from the other application's format to Unicode.

When you choose Paste As..., a “Convert From” dialog box appears, permitting you to select the code to convert from. Choose the appropriate one. If you do not know, just use trial and error, until the pasted text looks right.

Dictionaries in WinCALIS

User-Supplied On-line Dictionaries

Tools Menu Option "Look Up Word... F5"

You can type or highlight a word or phrase on the screen and look it up in an on-line dictionary, if one has been installed on your workstation.

If an on-line dictionary has been installed to accompany WinCALIS exercises, selecting the menu option Look Up Word... from the Tools Menu, or pressing the shortcut key F5, will open the On-line Dictionary dialog box, allowing you to look up the meaning or pronunciation of a selected word or phrase. Depending on how the dictionary was installed, it can be either accessible only from specific language keyboards (e.g. for a Korean-English dictionary, the current keyboard must be one of the Korean keyboards), or accessible from any language keyboard which does not have a specific dictionary linked to it (e.g. for a Korean-English dictionary, the current keyboard could be either the default English/W. Europe keyboard or any of the Korean keyboards).

A tiny sample dictionary of English has been included with the WinCALIS Demo exercises and Author Workstation Setup. It includes most of the vocabulary in the text "Turmoil at Grass Root Level" presented in the sample exercise "TURMOIL1.WCL." Try typing or marking a word like "keeper", "Manet", "arrondissement", "tax", or many other words. You can just mark the beginnings of words, especially inflected words, and WinCALIS will take care of the rest.

Full-scale bilingual dictionaries are also available or under development for various languages.

How to Look Up a Word

In WinCALIS you can type a word and look it up, or mark a word on the screen by dragging your mouse across it with the right mouse button depressed and look it up. You can also transfer the contents of an entire window to the WinCALIS Composition Editor and look up words from there. In the Composition Editor, WinCALIS Author, or UniEdit you can mark a word on the screen by dragging your mouse across it with the left button depressed, or double-click on a word and look it up.

You can press the Escape key at any time during a search to cancel further searching and display the matches found thus far. However, since the Escape key has a special meaning in WinCALIS (quit an exercise), Escape can not be used in WinCALIS for the first search, only from subsequent searches when the On-line dictionary dialog is already displayed.

In order to select a word to look up in WinCALIS, use one of the following three methods.

Method 1: Marking Directly in WinCALIS

- Holding down the right mouse button, drag the mouse across the word you wish to mark for looking up. The marked word will appear highlighted in a pop-up box in the lower part of the screen.
- Press the space bar or click the mouse in the box to close it.
- Press F5 or click on the menu item Tools, Look Up Word... to look up the marked word.

Method 2: Using the Dictionary Mode Window

- Copy the entire contents of a WinCALIS window directly to the "Dictionary Mode Window," by selecting the menu item Tools, Dictionary Mode Window.
- When the Select Window dialog box appears, select the window to be copied.

-
- Then the WinCALIS Composition Editor will automatically be launched with the clipboard text pasted into the Dictionary Mode Window, ready for looking up.
 - Drag the mouse across the word or phrase to be looked up (or double click on it if it is a single word) to highlight it.
 - Then press F5 or click on the menu item Tools, Look Up Word... in the Composition Editor to look up the word. Repeat as often as desired.
 - To return to WinCALIS, either exit the Composition Editor or leave it running for further use and switch to WinCALIS by using the Windows shortcut key Alt+Tab.

Method 3: Copying a Window to the Clipboard and Pasting in the Composition Editor

- Copy the entire contents of a WinCALIS window to the Windows clipboard by selecting the menu item Tools, Copy Window to Clipboard.
- When the Select Window dialog box appears, select the window to be copied.
- Then use Alt+Tab to switch to the WinCALIS Composition Editor (a special version of UniEdit). If this is not yet running, launch it by selecting the Composition Editor menu item from the Tools menu. Paste the clipboard contents into any document window.
- Then proceed as in Method 2 to drag the mouse across the word or phrase to be looked up (or double click on it if it is a single word) to highlight it.
- Then press F5 or click on the menu item Tools, Look Up Word... in the Composition Editor to look up the word. Repeat as often as desired.
- To return to WinCALIS, either exit the Composition Editor or leave it running for further use and switch to WinCALIS by using the Windows shortcut key Alt+Tab.

You can press the Escape key at any time during a search to cancel further searching and display the matches found thus far.

The On-line Dictionary Dialog Box

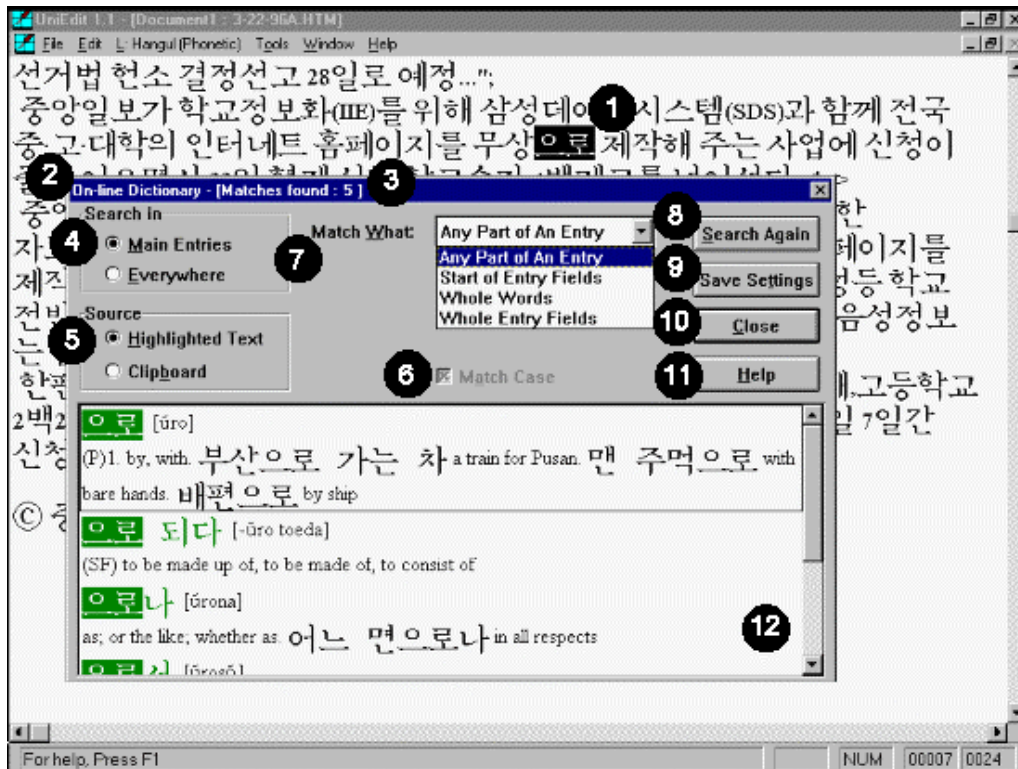


Figure 25: On-line Dictionary Dialog Box

The On-line Dictionary Dialog Box (2) contains various option settings in its upper part and a list box (12) displaying the items found matching the word looked up in its lower part. Options are explained below. The words in the dictionary matching the word looked up are highlighted in reverse video. The total number of matches (3) found is displayed in the title bar of the dialog box.

Options in the On-line Dictionary Dialog Box

Note: In order to understand the following options, note that an "entry" refers to an entire dictionary entry, which is enclosed in an outline box within the dictionary list box. You can scroll down through the displayed entries using the mouse and the scroll bar or using the PageDown, PageUp, and arrow keys. The term "field" is used to refer to the separate sections of a whole entry:

- the main entry field, highlighted in dark green, left justified on the first line of an entry
- the pronunciation or gender tag field, enclosed in square brackets, to the right on the first line
- the definition field(s), beginning on the second line of an entry

Escape Key

You can press the Escape key at any time during a search to cancel further searching and display the matches found thus far. However, since the Escape key has a special meaning in WinCALIS (quit an exercise), Escape can not be used in WinCALIS for the first search, only from subsequent searches when the On-line dictionary dialog is already displayed.

Radio Button and Checkbox Options

Search in (4):

- Main Entries

Displays only the matches found in the main entry portion (field) of a full entry. The main entry field is displayed in dark green at the left margin of the first line of a full entry. Main entries are what the user of a dictionary most commonly searches for. This is the system startup default option.

- Everywhere

Displays matches found in all fields of a full entry (displayed in dark green or in black).

Source (5):

- Highlighted Text

Uses the highlighted text as the word or phrase to look up. This is the system startup default option.

- Clipboard

Uses the Unicode text currently on the Windows clipboard as the word or phrase to look up. This feature is primarily designed as a way to use the WinCALIS on-line dictionary system with other Unicode-compliant applications. For looking up words from non-Unicode applications, use the "Paste As..." option in the Edit Menu of the WinCALIS Composition Editor, WinCALIS Author, or UniEdit and select the appropriate coding standard to convert from. Then highlight the word or phrase to look up in the normal way, using the default "Highlighted Text" as Source.

Match Case (6)

Will match lower case or upper case by default. If unchecked, will ignore case in finding matches. Since the WinCALIS system understands case relations in all the world's alphabets by consulting a table, case-sensitive searches are considerably faster than searches which ignore case.

Drop-down Listbox Options

Match What (7):

- Any Part of An Entry

The search will match any part of an entry, including phrases, words, and sequences of as few as three letters occurring anywhere in a word. This is the broadest search scope. This is useful when looking up words which are not likely to occur at the beginning of an entry, such as suffixes like English "-ment" or "-tion," or when using the dictionary "in reverse," to look up words which occur in definition fields. For example, in a Spanish-English dictionary, one would use this option along with "Search In: Everywhere" to look up the English word "house" in order to find its Spanish equivalent. It will find all cases where the word "house" occurs.

When looking up *hanja* Chinese characters in the Korean-English dictionary, this option should be used along with either the "Search in: Main Entries" option (will match only *hanja* main entries) or the "Search in: Everywhere" option (will also find any *hanja* occurring within the definition fields).

- Start of Entry Fields

The search will match the start of any entry field specified. For example, if "Main Entries" only are searched in (the default option), then "Start of Entry Fields" will match only the first letters of the Main Entry. This is the next broadest search scope and is the system startup default option. If all entry fields are searched in (the option "Everywhere"), then the "Start of Entry Fields" option will match the first letters of any field.

This is the best mode for handling language inflection, i.e., the variation of endings for verbs and nouns--highlight only the stem portion of a verb or noun to look up. This is also the typical search mode we use when looking up words in print dictionaries and glossaries. In other words, when we look up the English word "pruning," we scan with our eyes the main entries in a print dictionary for "prun...", until we find the appropriate match "prune," having regularized or "lemmatized" in our mind the "ing" form of "prune." The computer will not perform this lemmatization automatically, so that it will not find the match "prune," if the search string is "pruning." But it will find "prune" if the search string is "prun."

- **Whole Words**

The search will match any "whole word," which is defined as a sequence of letters bounded by spaces or punctuation, including the hyphen. This is the second narrowest search scope. Specific language dictionaries may also apply special principles. For example, in the WinCALIS Korean-English dictionary, bound forms have been tagged with an invisible Unicode Non-Spacing Joiner, which behaves like a regular character, so that bound forms are ignored during a "whole word" search.

- **Whole Entry Fields**

The search will succeed only if the word or phrase looked up matches an entire entry field precisely. This is the narrowest search scope. Matching whole entry fields has limited usefulness, but can be helpful, for example, in matching full main entry or pronunciation fields.

Button Options

Search Again (8)

Reinitiates a search using the current search options. It is often advisable during a lengthy search for too broad a search string to press the Escape key and narrow the search somewhat. For example the on-line dictionary search engine would find over 9000 matches for the search item "giv" in the Japanese-English dictionary, since "giv" is the marker for a Japanese given name.

Save Settings (9)

Retains the current search options for subsequent searches during the current WinCALIS session.

Close (10)

Closes the dialog box. You can also close the dialog box by pressing the Escape key.

Help (11)

Calls up On-line Help.

Japanese-and-English Dictionary

Key to Abbreviations

an.....adjectival nouns or quasi-adjectives (keiyodoshi)
a-no.....nouns which may take the genitive case particle "no"
vsnoun or participle which takes the aux. verb suru
vt.....transitive verb
vi.....intransitive verb

id	idiomatic expression
col	colloquialism
vul	vulgar expression or word
pn	person name (family or given)
pl	place name
giv	given name
fam	familiar language
pol	polite (teineigo) language
hum	humble (kenjougo) language
hon	honorific or respectful (sonkeigo) language
pref.....	prefix
suf	suffix
uk	word usually written using kana alone
uK	word usually written using kanji alone
oK	word containing out-dated kanji
io	irregular okurigana usage
arch	archaism
X	rude or vulgar term

The WinCALIS Japanese ↔ English on-line dictionary has been adapted to the WinCALIS on-line dictionary system and converted to Unicode/ISO 10646 from the "Jim Breen Japanese/English Dictionary." It is used with the permission of the copyright owner, Prof. Jim Breen of Monash University.

We are indebted to Prof. Breen not only for his permission to adapt and distribute his Japanese/English dictionary, but also because the simple but elegant architecture of his dictionary provided a model for our on-line dictionary and automatic indexing system in general.

Korean-and-English Dictionary

Key to Abbreviations

(ABBR).....	Abbreviation
(AD).....	Adverb
(ADJ)	Adjective
(BF).....	Bound form, bound noun
Ch.....	Chinese
(CJ)	Conjunction
Eng.	English
(HS).....	ROK Ministry of Education "High School" <i>hanja</i> (<i>hanja</i> learned by Korean high school students)
(IN)	Interjection (including onomatopoeia)
(MS).....	ROK Ministry of Education "Middle School" <i>hanja</i> (<i>hanja</i> learned by Korean middle school students)
(N).....	Noun
(NC).....	Numeral classifier (numerical counter, measure)
(Obs.)	Obsolete or archaic form
(PA).....	Particle, auxiliary word
(PF)	Prefix
(PN)	Proper noun (personal name, place name)
(PR).....	Pronoun
(SF)	Suffix
(VE)	Verb

(VC).....	Verb causative
(VG)	Verb gerund
(VI).....	Verb intransitive
(VN)	Verbal noun
(VT).....	Verb transitive

Key to Symbols



An arrow indicates a cross-reference, synonym, or other relevant usage.



Solid lenticular brackets enclose *hanja* Chinese character information in an "Okp'yon" character-dictionary-style field. See the item "Hanja Okp'yon Field" below for more information.



Hollow lenticular brackets enclose the variant inflected stem forms of common verbs.



Tortoise shell brackets enclose *hanja* Chinese character equivalents for Sino-Korean noun and bound morpheme (BF) entries.



A less-than sign means "(etymologically) derived from." Etymologies have been included for many recent loan words of European origin.



Parentheses enclose usage tags, meaning clarifications, and specialized terminology. E.g., "(Music)" indicates a term used especially in musical terminology; "(Medicine)" indicates a term used especially in medical terminology; and "(Obs.)" indicates an obsolete or archaic form.

General Features

Layout

A sample dictionary entry is pictured below, with labels for each field. See below for further details on each field and on other dictionary options.

시 (市) [shi]
(N) city, town; (BF) market, fair; trade; execution ground 【저사市 MS】

Figure 26: Sample Korean - English Dictionary Entry

Main entry field

Main entries are highlighted in dark green. For verbs the main entry includes the dictionary citation form, as well as variant inflected forms for the most common and short verbs, enclosed in hollow lenticular brackets. The purpose of this is both for reference and to make it easier to locate a verb by highlighting and looking up its stem element. For Sino-Korean nouns, the *hanja* Chinese character equivalent is given, enclosed in tortoise shell brackets.

Pronunciation field

The second field of each entry, after the main entry on the same line, is the pronunciation field, enclosed in square brackets. Pronunciation is given according to the McCune-Reischauer pronunciation.

Alternate pronunciations are provided within the definition field, preceded by "Also":
Note example entry:

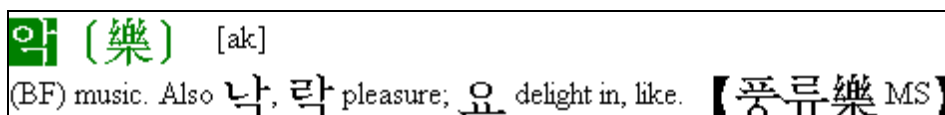


Figure 27: Example Dictionary Entry, using "Also"

Definition field

Definitions in English, derived forms (e.g., adjectives or verbs formed from nouns), examples with English translations, and usage information are all presented in the definition field, beginning on the line following the main entry field. Distinct definitions are grouped under separate Arabic numerals, in accord with common Korean lexicographic practice. For most verbal nouns ending with *hada*, the noun is given as the main entry, and the *hada* form is given within the definition field.

Hanja "Okp'yon" field

The entries for the most common morphemes of Chinese origin end with a *hanja* "Okp'yon" Chinese character dictionary-style field, enclosed in solid lenticular brackets. This usually contains first a Korean "tag meaning" for the morpheme (*hanja* character). Second is the *hanja* character itself. Third is an approximate indication of frequency of usage, if it is one of the 1800 characters designated as the most important for educational purposes by the Ministry of Education of the Republic of Korea. "MS" indicates that it is one of the most common 900 characters which are taught in middle school in the Republic of Korea. HS indicates the next most common 900 characters, which are taught in high school in the Republic of Korea.

Serial Arrangement

Entries are arranged in ascending *hangul* alphabetical order, in the order of the Unicode/ISO 10646 encoding standard. This order is in turn based on the Korean Standard KS C 5601-1989.

For more information on Unicode, consult The Unicode Consortium, *The Unicode Standard: Worldwide Character Encoding*. Version 2.0 (Reading, Mass: Addison-Wesley Developers Press, 1996); for more information on ISO 10646, consult the document *ISO/IEC 10646-1:1993 Part 1: Architecture and Basic Multilingual Plane* (available from ANSI, 11 West 42nd St., New York, NY 10036).

Note: During 1996 the Unicode/ISO 10646 code points for all Korean *hangul* underwent a shift from their location in the code range 3400-4D00 under the Unicode 1.1 standard to their new location in the code range AC00-D7A3 in the revised standard Unicode 2.0/ISO10646. In order to both accommodate the previous standard and anticipate the revised standard within the WinCALIS Korean-English dictionary, the dictionary lookup engine has been designed to handle *either* range of code points. The dictionary is distributed in two different versions, one for Unicode 1.1 and one for Unicode 2.0. Either can be installed on the user's system, or both can be installed simultaneously, if they are linked to one or more separate language keyboards under the Language menu--for example the existing "Hangul (Phonetic)" keyboard can be linked to the dictionary using the Unicode 2.0 standard, while a new "Hangul (Phonetic) Alt." keyboard can be linked to the dictionary using the Unicode 1.1 standard. In order to access a specific dictionary, then, the user would need to switch keyboards to the desired one before looking up a word. In theory, these two dictionaries could be combined and reindexed together, so as to

allow looking up Korean *hangul* encoded in *either* Unicode 1.1 or Unicode 2.0, but this not only would produce a dictionary double the current size, but also would have the undesirable effect of displaying duplicate entries for all *hanja* and English words looked up. It would also be more difficult to maintain and extend.

Grammar

The system of grammar employed is generally that of Fred Lukoff, as found in his textbook, *An Introductory Course in Korean* (2nd edition, Seoul:Yonsei University Press, 1993). The monumental dictionary *A Korean-English Dictionary* by Samuel E. Martin et al. (New Haven: Yale University Press, 1967) was also extensively consulted, as well as most of the common Korean-English dictionaries published in Korea for learners of English.

Scope

The dictionary contains about 24,000 main lexical entries, with several thousand more derived forms given in the body of the entries. Pains have been taken to ensure that virtually all the vocabulary items presented in the Korean textbooks commonly used in the United States have been incorporated here. Other special areas of coverage which have been incorporated include:

- over four hundred lexical items which are unique to North Korean usage, with equivalent South Korean usage added as a cross-reference; indicated with the usage tag "(N. Korea)"
- common geographic names, both Korean and foreign
- common Korean surnames (marked "surname")
- common Chinese surnames (marked "Ch. surname")

Entries have been provided for all of the 4888 *hanja* Chinese characters from the Korean Standard KS C 5601-1989, plus others not included in the KS standard which are used to write common vocabulary. Each entry gives the *hangul* reading and McCune-Reischauer transcription. Many of these *hanja* have obscure meanings or no meaning at all when used independently, so in order to avoid distorting their function, they have been simply marked in the definition field with the phrase "[rare combining form]." However, it is hoped that the pronunciation provided will be of value in identifying and reading them when they occur in the names of places and persons.

Acknowledgments:

We are very grateful to the following teachers and scholars for their advice in the design and compilation of this Korean-English dictionary: Mrs. Yang S. Camus, Prof. Kay Kang, Mrs. Young Ah Lee (Chief Editor), Mr. Bo Park, Dr. Bruce Phinney, Prof. Han Sohn, Prof. Janna Son-Yarborough, and Mr. Sukmoon Yoon.

How to Create Your Own On-line Dictionary

The item "Look Up Word..." in the Tools menu is a powerful new feature in WinCALIS. Depending on what is installed on a particular workstation, WinCALIS, WinCALIS Author, or UniEdit (the WinCALIS Multilingual Text Editor) can access an unlimited number of Unicode-text on-line dictionaries. These dictionaries can be either multilingual (e.g., Russian-Arabic, Arabic-Russian), or monolingual (e.g., a dictionary of specialized vocabulary for advanced students of English).

Create a dictionary text file

Any author can generate a dictionary by creating a (Unicode) text file in WinCALIS Author or UniEdit in a fixed format, like:

EntryField/Pron.orGenderField/UsageField/DefinitionField1....

There should be one entry per line, separated by hard returns, and with fields separated by slashes (/). Note that because the slash has a special meaning, it is a reserved character and should not be used within fields. The EntryField will start a new entry and will be highlighted in dark green color. The second field ("Pron.orGenderField") will be placed on the same line surrounded by square brackets. If the second field is empty—that is, there are two successive slashes (/), the brackets will be replaced by a colon (:). Succeeding fields will be placed on new lines. The maximum length for an entire entry is practically speaking about 512 characters, since that is about how much can be viewed in the on-line dictionary list box at one time.

When you have created a small pilot dictionary file in a text window, save the file to disk in the ..\WINCALIS\DICTIONARIES subdirectory (or whatever path you have specified in the "Dictionaries=" line of WINCALIS.INI), with the file extender ".DIC", specifying the file type "Unicode/No Header" in the "Save File as Type:" box.

Generate the Dictionary Index

Then you need to generate the index to your dictionary. In WinCALIS Author or UniEdit, in the Tools menu, click on "Generate Index" and specify the name of the dictionary to generate the index for. Usually you should specify the same 1-8 letter filename for both the dictionary and the index to be generated. E.g., if your dictionary file is named "wcrusfrn.dic," then the index should be named "wcrusfrn.idx."

Associate with a Keyboard

At this time you can also select a language keyboard to associate the new dictionary with, if you wish to use more than one language dictionary. When that keyboard is the active keyboard in WinCALIS, WinCALIS Author, or UniEdit, this dictionary will be the one which is used. WinCALIS Author or UniEdit will create or update a [Dictionary] section in the corresponding language-keyboard .INI file on your workstation. Anytime after you restart WinCALIS and WinCALIS Author, or UniEdit, if that language-keyboard is the current keyboard, the programs will look for your dictionary and index file in the ..\WINCALIS\DICTIONARIES directory or other directory specified in the "Dictionaries=" line of WINCALIS.INI.

Note: In order to associate your dictionary with more than one keyboard, you may use the Windows Notepad to open the .INI file of the keyboard you first selected (check the time/date stamp on the file in the File Manager to determine which one it is, if you can't spot it), then copy the entire [Dictionary] section to the Clipboard and paste it into as many other .INI files as you wish. (See LATIN1.INI for a sample.) See also Language Keyboard .INI Files.

When you press the "OK" button in the Generate Index dialog box, Author or UniEdit will begin to create the index file. For a small pilot dictionary of just a few items, this process takes only a few seconds or minutes. But for a real glossary or dictionary of thousands of entries, and with many items to index, expect the index generation process to take 1-3 hours, even on a fast machine. It generally works well to start the index generator at the end of a day and let it run during the night. Also, be sure that there is plenty of free space on the hard drive for the several hundred temporary files which are created during the indexing process—at least 2-3 times the total size of the dictionary file itself.

Every alphabetic sequence of 3 letters will be fully indexed, as will every single Han character for Chinese-Japanese-Korean-Vietnamese. Note that at the moment there is no way to index 2-letter words such as English "to" or "by." There is also not yet a "stop-word" function, which will exclude certain 3-letter combinations, like "and," from being indexed.

When index generation is completed, the dictionary and index files are finished and ready to use. If you wish students to have access to your dictionary, you will need to arrange to install the dictionary files on the student workstation(s) and link them to keyboards as needed. If you would like to contribute your dictionary as a general enhancement to the WinCALIS system, we can pre-install your dictionary in student workstations.

Dictionary Creation Notes

Note that the default dictionary and index names are WINCALIS.DIC and WINCALIS.IDX. A dictionary with that name will be found regardless of the current keyboard, unless another dictionary is specified for the current keyboard.

Users who are developing large dictionaries should review editing limits imposed by the Author/UniEdit editor. In particular, note that there is a limit of 6000 lines of text in each window in WinCALIS Author and UniEdit. There can be up to six concurrent 6000-line texts, one in each window. You will need to develop your dictionary in component parts of ca. 5000 lines each. Then you can use the DOS COPY command or a DOS batch file to concatenate each of the component files into one big file (hence the "Unicode/No-Header" file type). There is a sample batch file, UPDATE.BAT, included in the ..\WINCALIS\DICTIONARIES directory during setup.

The dictionary file does not need to be in any particular order, and sorting it (with the "Sort Lines" item under the "Tools" menu) is useful mostly to avoid duplicate entries and so that display of items in the dictionary is in ascending order, when the word looked up is not a main entry. You can sort multiple component files in the same way you might sort a stack of cards to be alphabetized by first dividing it into piles by initial letter, then proceeding to alphabetize each of the piles, then recombining all the piles in order. You would proceed by sorting each of your files, then chopping each of them into eighth-length files (or quarters or halves or 1/20ths, depending on the total length of your dictionary--you will need as many divisions as you have component files) at logical divisions (e.g. in the Latin alphabet, a-c, d-f, g-i, j-l, m-o, p-r, s-u, v-z, then combining all the first-eighth files and resorting them; then all the second-eighth files and resorting; and so on. When all the sub-files are reassembled and sorted, they can be reassembled. The entire dictionary should now be in sorted Unicode order. A sample DOS batch file to do the concatenating operation mechanically is included in the ..\WINCALIS\DICTIONARIES subdirectory. Be sure that you keep master copies of all component sub-files--remember that full dictionary files exceeding 6000 lines are not directly editable or even viewable in the WinCALIS editor. If you accidentally combine all your component files and fail to keep the components, you can, in a pinch, use a file utility such as "SPLIT49.EXE" to split the big file again at even byte boundaries into any number of components. SPLIT49.EXE and its documentation, SPLIT49.DOC can be downloaded from the Humanities Computing Laboratory website, <http://www.humancomp.org>.

We are in the process of developing large bilingual dictionaries for Korean-and-English and Japanese-and-English. We hope that with the collaboration of interested WinCALIS authors and users, we will be able to distribute useful dictionaries to accompany WinCALIS for a wide range of languages.

Tools Menu Option "Generate Index"

This menu option in WinCALIS Author or UniEdit runs the automatic Unicode index builder to build an index for a dictionary file in the default "Dictionaries" subdirectory. The dictionary should be saved as type "Unicode/No Header".

Record-Keeping in WinCALIS

Record-keeping is a way to keep track of student use of WinCALIS. Besides seeing who completed a lesson and when, record-keeping can also capture student input, and reveal how long each task took to complete. WinCALIS keeps records in Borland Paradox table format.

The “RECORDS” subdirectory can be used to hold the Borland Paradox database tables which are generated by WinCALIS 2.x to hold student records, when record-keeping is installed.

The path to the “RECORDS” subdirectory is defined in the WINCALIS.INI file in the main WinCALIS 2.x subdirectory. It can be redefined to whatever path you choose by carefully editing the line

Records= ...

in the WINCALIS.INI file with a plain text editor, such as the Windows Notepad or the DOS “EDIT” program.

How to Install Record-Keeping with Paradox Tables

WinCALIS maintains varying levels of record-keeping to capture student action and responses. (In addition to the <REPORT> and <QUIZ> features). The following section is a brief description of how to implement and use this record-keeping information.

Setting Up

Modifying the WinCALIS.ini file

Record-keeping is not enabled in a regular WinCALIS setup. To enable it, open and edit the “WinCALIS.ini” file for the installation of WinCALIS where you want student input to be recorded (at a certain machine, or in a certain lab),. Locate the line that begins “Records=”. This line must point to the directory where you want the record-keeping information to be accessed. Uncomment the Records= line (delete the semicolon and then save the file).

Excerpt from sample WinCALIS.ini file:

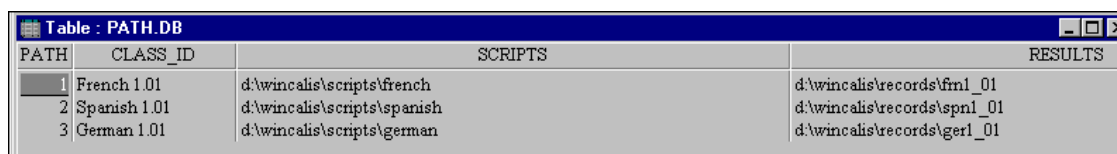
```
[WinCALIS]
Home=d:\WINCALIS\
Scripts=d:\WINCALIS\SCRIPTS
Graphics=d:\WINCALIS\GRAPHICS
Multimedia=d:\WINCALIS\MLTMEDIA
Records=d:\WINCALIS\RECORDS
Conversions=d:\WINCALIS\CONVERT
Documents=d:\WINCALIS\DOCS
Dictionaries=d:\WINCALIS\DICTIONARIES
GlobalInfoFile=d:\WINCALIS\WinCALIS.wgi
SubMenu1=Arabic script
SubMenu2=African
SubMenu3=Asian (other)
```

Providing class, student, and directory information

In Paradox, or another database program that can handle *.db files, open the file called “path.db”. This identifies the following fields:

CLASS ID (Name of class, referenced also in studrec.db file)
 SCRIPTS (directory path indicating where scripts are located)
 RESULTS (directory path indicating where the record files are to be placed)

Enter your class names, scripts path and results paths here. If you point to a new subdirectory in a RESULTS path, be sure to create such a subdirectory in a file management program (in the example, there is a frn1_01 subdirectory for records from the French class).



PATH	CLASS_ID	SCRIPTS	RESULTS
1	French 1.01	d:\wincalis\scripts\french	d:\wincalis\records\frn1_01
2	Spanish 1.01	d:\wincalis\scripts\spanish	d:\wincalis\records\spn1_01
3	German 1.01	d:\wincalis\scripts\german	d:\wincalis\records\ger1_01

Figure 28: Sample path.db file

Make modifications and additions as needed for your system, and then save this file with the same path.db file name.

Initial setup information must also be entered in the studrec.db file. The fields in this file are:

LOGIN (Max. 8-character identification for each student - Case sensitive)
 PASSWORD (Max. 20-character password)
 CLASS_ID (These must correspond directly to the entries in the same field of the “path.db” table)
 LOG (Refers to level of information: 1 or 2)
 STUDENT_NAME (Space to write out full name of each student)

STUDENT	LOGIN_ID	PASSWORD	CLASS_ID	LOG	STUDENT_NAME
1	jane		French 1.01	1	Jane Doe
2	4367		French 1.01	2	Roy Schwartz
3	minwang		Spanish 1.01	1	Min Wang
4	doggie		German 1.01	2	Doggie Dog
5	sls		French 1.01	2	Satsuki Scoville
6	rick		Japanese 201	2	Rick Rack

Figure 29: Sample studrec.db file

Edit a file that reflects your students and classes and the level at which you wish to have records. Save this file as “studrec.db”.

For the basic level of record-keeping, recognized as LOG 1, then WinCALIS is writing to a file that is automatically named *****r.db (where the asterisks are the first seven characters of the lesson. Level 1 makes a *record* of student input). For the higher level, recognized as LOG 2, information is being written to *****c.db. Level 2 makes a *capture* of student input. These files will be placed in the directory that was indicated in the RESULTS field of the path.db.

When WinCALIS is launched, the user is invited to enter a login name and password. If this user is found in the “studrec.db” file AND the lesson that he or she runs is in “path.db” file, then WinCALIS is monitoring input during that particular lesson. Once the student has completed the lesson, then the records file will be placed in the results directory.

Looking at Records:

Once the files are arranged correctly, and some lessons have been attempted, then you can view the records. Look in the directory that you commanded the results to go, and look for the .db file beginning with the first seven letters of the lesson name that you want to analyze. For those that you have indicated as Level 1, there will be an "r" after the name. For those looking at Level 2, there will be a "c" after the name.

01-3AREC	LOGIN_ID	SCRIPT_ATTEMPT_NUM	DATE	CORRECT	ATTEMPTED
1	cindys	1	9/11/95	10	10
2	crafton	1	9/13/95	9	10
3	hanuman	1	9/13/95	6	6
4	hanuman	2	9/13/95	6	6
5	jarnold	1	9/12/95	9	9
6	jarnold	2	9/12/95	9	9
7	jarnold	3	9/13/95	10	10
8	mag	1	9/6/95	10	10
9	mooner	1	9/12/95	6	6
10	teddy	1	9/9/95	9	10
11	wyntonma	1	8/30/95	0	1
12	wyntonma	2	9/11/95	9	9

Figure 30: Sample of a WinCALIS Level 1 records file

RECRDCAP	LOGIN_ID	SCRIPT_ATTEMP	TASK_LABEL	TASK_ATTEM	RESP_TYPE	RESPONSE	RESP_TIME
1	rick	3	1	1	Incorrect	Clicked on "False"	4,408.00
2	rick	3	1	2	Correct	Clicked on "True"	206.00
3	rick	3	2	1	Correct	Clicked on "d. 3"	1,428.00
4	rick	3	3	1	Incorrect	sunday	7,040.00
5	rick	3	3	2	Incorrect	friday	2,329.00
6	rick	3	3	3	Correct	Friday	1,455.00

Figure 31: Sample of a WinCALIS Level 2 records file

Details:

--Records on student 'rick'

--This is the third time that Rick attempted this lesson

--At Level 2, WinCALIS reports the number of times a person attempted to respond to each task, as well as all input on all attempts, and time elapsed for each attempt.

If you or your school owns or plans to order Borland Paradox, then we will provide you with a set of forms that were designed for use with the WinCALIS records tables. Contact the Humanities Computing Laboratory for information.

Record-Keeping Troubleshooting:

<i>PROBLEM:</i>	<i>POSSIBLE REASON:</i>	<i>SUGGESTION:</i>
“Table was not found” message	WinCALIS could not locate the necessary files: studrec.db and path.db.	Check for or create those files as specified above.
“Record was not found” message	WinCALIS could not locate login name	Check for student login identification name and password.
“Cannot write to directory” message	WinCALIS cannot find the scripts path or the results path.	Check to see that the directory paths are correct in path.db and that all directories listed actually exist (if you point to a new subdirectory, then create a new subdirectory in a file management program).
“Multiple Paradox net files found” message	WinCALIS cannot write to an open file	Close studrec.db and path.db files.
No student login prompt	Record-keeping is not enabled	Edit WinCALIS.ini file and uncomment the lines that begin “Records=”

WinCALIS Record-Keeping: Viewing Through Paradox Forms

WinCALIS’s record-keeping feature will capture various parts of a student’s response to WinCALIS exercises. It stores these responses in Paradox format tables, which can be opened in Paradox to analyze how well a student or class is doing on the exercises.

Before you employ the Paradox forms, be sure that you have properly set up the record-keeping option and have been using it successfully. The purpose of this section is to explain the use of one form that has been created to facilitate the viewing and analysis of the response tables. This particular form is very rudimentary and is limited in efficiency by certain WinCALIS quirks that will be improved in subsequent releases. This form is not the only way to analyze WinCALIS’s response tables; it is merely an example of one possible approach. For instance, support for viewing captured student responses to tasks is not included in these forms. Any user who is well versed in Paradox may create other forms which may better suit their needs.

Installation

The record-keeping tables that WinCALIS uses are PATH.DB, STUDREC.DB, and one or two other tables for each exercise executed with record-keeping engaged. The location of all of these tables must be known beforehand in order to use the form. PATH.DB and STUDREC.DB are usually found in the RECORDS subdirectory of the WinCALIS directory, and the response tables are usually found in subdirectories of the RECORDS directory, depending on the setup.

To install the forms, obtain the appropriate ZIP file containing the forms. A new directory must be created in a convenient place, it doesn’t matter where so long as it is empty. Unzip the ZIP file into this directory. It should contain the following files: EXRESULT.FSL, EXRESULT.RSL, RECORDS.FSL, STARTUP.FSL, STRESULT.FSL, and STRESULT.RSL. Start Paradox and set the working directory to the directory containing these forms.

Note: IMPORTANT: WinCALIS must not be accessing the record-keeping tables while the form is running or it will not function properly. WinCALIS accesses the tables when a student logs in and when a student finishes a question or exercise, depending on the level of capturing (if capturing is engaged). When in doubt, make sure no one has the system copy of WinCALIS open at all.

The main form is RECORDS.FSL, however certain tables and variables must be created before the form can run. In the Paradox Project Viewer, click on Forms and then double-click on STARTUP.FSL, and the forms will initialize properly.

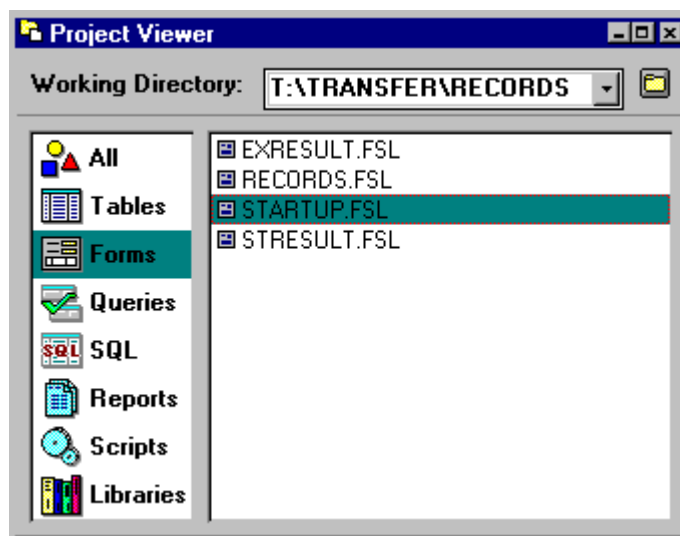


Figure 32: Choose 'startup.fsl' for the forms that view records tables

STARTUP must always be run before RECORDS, not just the first time. The first time the forms are run, STARTUP.FSL will prompt the user for the location of PATH.DB and STUDREC.DB. The location of these files will be stored in a text file called RECORDS.DIR. This file may be deleted whenever the user wishes. STARTUP will create it again if it cannot be found. This file must be deleted or edited if PATH and STUDREC change locations.

Because WinCALIS currently does not allow STUDREC.DB and PATH.DB to be indexed (this will be changed in the future) it is necessary to make a copy of these tables in order to establish an index to link the tables. STARTUP automatically copies these tables to PATHCOPY.DB and SRECCOPY.DB. The user should be aware of this because any changes made to these tables while using the form will only affect the copies of these two tables until the changes are saved. So remember to save changes before exiting!

STARTUP will also create a table called EXLIST.DB, which will be used to list all exercises used for analysis, and RESULTS.DB, which will tabulate the results for each exercise in EXLIST. A table called NUMPROBS will also be created to allow the user to specify the number of tasks in each exercise, so scores can be calculated as percentages of the total.

Once STARTUP has run and created all necessary tables, it will automatically load RECORDS, which looks like this:

WinCALIS Student Response Statistics

1 Analyze by Exercise (selected) 2 < 3 < Prev. 4 Next > 5 > 6 Save Changes 7 Revert 8 Exit

9 Preview Results 10 Print Results

11 Selected Exercises 12 Class ID : French 1.01 13 Read Directory 14 Remove from List 15 Add to List

16 Scripts Dir : n:\winprgs\wincalis\scripts\french 17 Results Dir : n:\winprgs\wincalis\records\frn1_01

Student Name	Login ID	Password	Capture
Erica Lee Gotham	elg3	batman	2
Maria Consuelo Rodriguez	mcr1	happy	2
Roy Schwartz	rs5	hwdooddy	2
Prof. Charles Sinclair	sinclair	applebite	1

Selected Exercises	# of Tasks
BABETT2C.DB	7
BABETT2R.DB	7
PREPFREC.DE	18
PREPFRER.DE	18

18

Figure 33: Viewing WinCALIS records tables through Paradox form

It may look pretty complicated, but it's not hard to use. The form is organized by class. Scrolling through each class displays the class's path information in the "Scripts Dir" and "Results Dir" fields, the student's information for the class in the bottom table window, and the exercises selected for analysis in the far right table window. An explanation of the controls follows.

Controls

- 1: This radio-button box determines how the results should be organized. If "Analyze by Exercise" is selected, the exercises will be stepped through one by one, and the students' scores will be listed under each. If "Analyze by Student" is selected, the students will be stepped through one by one, and the results of each exercise they have attempted will be displayed.
- 2: Moves to the very first class in PATH.DB.
- 3: Prev: Moves to the previous class listed in PATH.DB.
- 4: Next: Moves to the next class listed in PATH.DB.
- 5: Moves to the very last class listed in PATH.DB.
- 6: Save Changes: Posts whatever changes have been made to class path and student information to STUDREC.DB and PATH.DB. If this is not executed before the form is closed, changes will be lost.
- 7: Revert: Discards whatever changes have been made to class path and student information and reloads the records in STUDREC.DB and PATH.DB.
- 8: Exit: Prompts to save changes then exits the form.
- 9: Preview Results: Calls another form to display on the screen the results of the selected exercises.

-
- 10: Print Results: Prints a hard copy of the results of the selected exercises.
 - 11: This table window displays the list of exercises that have been selected for analysis (the exercises that are listed in EXLIST.DB). In the “# of Tasks” column, the number of tasks each exercise contains may be specified so that scores may be calculated as a percentage of this total. It is not required that an exercise have a value for this field.
 - 12: Class ID: Displays the currently selected class. This field can be edited to change the name of a class, or a new class can be inserted into this field.
 - 13: Read Directory: This looks in the directory specified in the “Results Dir” field for exercise result tables that are named “*R.DB” or “*C.DB.” If any are found, their names are stored in EXLIST.DB and so they will appear in the far right table window.
 - 14: Remove from List: This takes the user to a file browser. Any files the user selects will be removed from the selected exercise list. Please note that only files in the “Results Dir” directory can be specified; directory changes in the file browser will result in an error message.
 - 15: Add to List: This also takes the user to a file browser. Any files the user selects will be added to the selected exercise list. Again note that only files in the “Results Dir” directory can be specified.
 - 16: Scripts Dir: This field displays the path to the selected class’s scripts. This field cannot be edited directly- the “Scripts Dir” button must be pressed, which will take the user to a file browser to select a new directory.
 - 17: Results Dir: This field displays the path to the selected class’s response tables. This field cannot be edited directly- the “Results Dir” button must be pressed, which will take the user to a file browser to select a new directory.
 - 18: This table window displays information on all the students in the selected class. Each of the fields can be edited, and new students can be inserted here. Please note that the only acceptable values for the “Capture” field are 1 and 2: one to capture only the students’ scores, and two to capture students’ responses to the tasks.

Example

As an example of this form’s use, suppose that there was a class listed in PATH.DB called “French 1.01.” To find out how the class has done on their exercises, click on the appropriate arrow buttons to move to this class, then click on the “Read Directory” button. All response tables found for the class in the results directory will appear on the exercise list.

Now suppose one of the exercises in the list was an optional assignment; click on “Remove from List” to exclude it from the analysis. If the exercise needs to be included after all, click on “Add to List” and select the exercise to add it back.

If the results should be grouped by students, be sure that “Analyze by Student” is checked. To group the results by individual exercises, check “Analyze by Exercise.” Then click on either “Preview Results” or “Print Results” and the students’ scores will appear on the screen or at the printer.

Editing Paradox Records

It was previously mentioned that values listed in the fields on the form could be changed. To do so in Paradox, Edit Mode must be engaged by pressing F9. To edit a field, press F9 and click on that field and make appropriate changes, and then press F9 again to exit edit mode and post the changes to the table.

Note: IMPORTANT: Always remember to exit Edit Mode before performing any actions on the form. If Edit Mode is still engaged and the record is not posted to the table, any action performed on the form, say closing it, will function as if the record hasn’t been changed, and in this case the changes would be lost.

Note: If you want to insert a new record, and so press F9 and overwrite a field's value, you are changing the value of that record and not creating a new one. To insert a new record in Paradox, press the Insert key and a blank record will be displayed for editing.

Previewing the Results

Depending on whether the results are being grouped by student or by exercise, when the "Preview Results" button is clicked, one of two forms will load to display the results. On these forms, there will be arrow buttons similar to those on the RECORDS form for stepping through the records, and a "Close" button to exit the preview form and return to RECORDS. There will also be a "Save to File" button. Both of the forms are based on the same table (RESULTS.DB). The "Save to File" button allows the user to copy the RESULTS table to a new name and location in case permanent electronic record of the results is desired.

Deleting Files

These forms create a lot of tables when used. If the user wishes to conserve disk space, the tables created by the forms may be deleted when the forms are not running. They will be created again the next time the forms are run. (Just be sure not to delete PATH.DB and STUDREC.DB.)

The best way to delete a table is within Paradox. Tables are not just single files; other files are also associated with the table, such as indexes and saved viewing properties. If tables are deleted within the Paradox Project Viewer, all files associated with the table are sure to be removed, not just the *.DB file. To do this, click on "Tables" in the Project Viewer and then right-click on the table you wish to delete. A list of options will appear, from which you can choose to delete the table.

Note: EXLIST.DB and RESULTS.DB are created in the private directory. This was done to ensure proper sharing of files over a network- no other user should share the same EXLIST.DB and RESULTS.DB files. Make sure that if multiple copies of Paradox are running over a network, no two users have the same private directory. (This applies for most Paradox databases, not just this particular form.)

Printing a WinCALIS Script or Document

You can use WinCALIS printing facilities to print a *.CAL script or any document generated in the Scratchpad or other window and saved as a text file on disk. The WinCALIS Author Workstation package comes with two printer programs, the WinCALIS Multilingual Printer program (MLP.EXE), for text in any language and script, and the WinCALIS TrueType Printer Program (MLPTT.EXE), for text written in WinCALIS Author entirely using a scalable TrueType font. (Currently WinCALIS does not support TrueType fonts for Chinese, Japanese, Korean, a few other scripts such as Amharic, and miscellaneous symbol and exotic Roman character fonts.)

If your text is by default written in a TrueType font (most alphabetic languages), you are advised to print with the TrueType Multilingual Printer Program (MLPTT.EXE), which is faster, more powerful, and produces superior-quality printing at any font size. You can start this program by clicking on its icon in the WinCALIS Pangaea group. If you are not sure, experiment by printing a short test document with the TrueType Printer Program. Non-TrueType characters will merely not display properly or at all on the printed page. If this affects your print job, you should resort to printing with the Multilingual Printer.

WinCALIS Multilingual Printer Program (MLP.EXE).

Selecting the File,Document file,Print menu item in WinCALIS Author launches the WinCALIS Multilingual Printer program (MLP.EXE). Because the printing process takes a snapshot of every page it sends to the printer, the process is a bit slow. However, this method allows WinCALIS Author to work with virtually any printer, and enables you to generate printed material in any language supported by WinCALIS.

When you choose Print (Alt+F,P) from the File menu a dialog box opens. Within the Multilingual Printing dialog box, you will be presented with the following menu items:

- Settings
- Print
- Printer Setup
- Print Help

Settings

The default settings for margins, paper size, and page orientation are set so you can produce attractive documents without having to alter the settings.

Margins

The settings you choose for margins affect all pages of a document. The default settings for margins are fixed so you can produce attractive documents without having to change the settings.

Margin settings determine the point at which text appears on a page. If you want to emphasize certain paragraphs, you can indent them to set them off from the margins. You can change the script's length and improve readability of a document by altering its margin settings.

Type the measurements for the margins you want to adjust: Top, Bottom, Left and Right. The measurements must be set in inches.

Header

Prints the filename as a header at the top of each page.

Text

You can use Text formatting to align text and adjust spacing in your script. You can apply Font Size, Line Spacing, and Kerning.

Font Size

Choosing the appropriate font size for text can set the tone for a document and affect its readability by varying the pixel size in your script. Your selection of sizes is determined by your printer.

Line Spacing

The default setting of 0.02 inches prints text approximately as spaced on the screen. However, you can increase or decrease the amount of white space between lines in a document.

Kerning

You can change the spacing or kerning of characters to make the script more readable.

Print Scaling

You can change the size of the font in which your document is printed at a given printer resolution, such as 150 dots per inch, from teeny-tiny (Print Scaling 1 -- i.e., 1 pixel on the screen is printed as 1 dot on the paper), to medium (Print Scaling 2 -- i.e., 1 pixel on the screen is doubled both horizontally and vertically as 4 dots on the paper), to magnified in size, but not quality, through further multiplication of pixels (Print Scaling 3 and above). Default is Print Scaling 2.

Print

Using the Print option you may print any document file.

To print a WinCALIS *.CAL script or text generated in the Scratchpad:

- From the Author File menu, choose Print (Alt+F,P) to open the Multilingual Printer dialog box. Or start the program by clicking on its icon in the WinCALIS Pangaea group.
- Next, in the Print menu find and select the file you wish to print.
- Then press the Print button to print the document or press the Preview button to view the document before printing.

Printer Setup

The Multilingual Printer will print on the printer installed as the Windows default printer. If more than one printer is installed, use the Windows Control Panel, Printers “applet” to set the desired printer as the default printer. Printing options are printer-specific settings, including resolution, color, orientation, text quality, and others. For additional information on changing the settings, consult your Windows manual.

WinCALIS TrueType Printer Program (MLPTT.EXE)

All of the WinCALIS fonts can be printed using the WinCALIS Multilingual Printer (MLP.EXE). However, WinCALIS also supports several alphabets in TrueType fonts. Thus when printing text as TrueType, you may want to try the TrueType Multilingual Printer (MLPTT.EXE). You may use the TrueType Printer for alphabets such as Arabic, Armenian, Greek, Georgian, Hebrew, and Thai, well as

certain other alphabetic languages. To receive updates on character sets, please contact Humanities Computing Laboratory or its agents.

The TrueType Printer program also provides a handy place for quickly opening and viewing WinCALIS Unicode files of virtually unlimited length. And since the file is opened in read-only mode, it has some of the speed and safety of other file viewers for DOS and Windows, like “LIST” or XTree.

When you choose Print (Alt+F,P) from the TrueType Printer File menu a dialog box opens. Within the printing dialog box, you will be presented with the following menu items:

Print
Print Preview
Print Setup
Print Settings

Print...

Using the Print menu option you may print any document file.

- **To print a WinCALIS *.CAL script or text generated in the Scratchpad:**
 - Start the program by clicking on its icon in the WinCALIS Pangaea group.
 - From the File menu, choose Print (Alt+F,P) to open a dialog box.
 - Next, using the Open option in the File menu, open the file you wish to print.
 - From the File menu, choose Print to print the document or Print Preview to view the document before printing.

Print Preview

Print Preview shows an entire page at a reduced size. You can check page breaks, position of text, and layout. When you choose Print Preview (Alt+F,P) from the File menu, WinCALIS printing module displays a Document dialog box with the following menu items:

Print	Prints the document.
Next Page	Displays the next page.
Previous Page	Displays the previous page.
Two Page Zoom In	Displays two pages at once. Controls portion of the page you see in your document window. Allows you to magnify part of a page.
Zoom Out	Controls portion of the page you see in your document window. Allows you to reduce part of a page.
Close	Closes the File menu and returns you to the Multilingual Printing dialog box.

Print Setup...

Printing options are printer-specific settings, including resolution, color, orientation, text quality, and others. For additional information on changing the settings, consult your Windows manual.

Print Settings...

The default settings for margins, paper size, and page orientation are set so you can produce attractive documents without having to alter the settings.

Chapter 8:

New Features in WinCALIS

New WinCALIS Features and Extensions to CALIScript

Creating a BUTTON

Buttons

New commands

<BUTTON name CREATE,"Label" [NORMAL w,h]>

Create button 'name' with ANSI Label" (max. 55 characters) [w pixels wide, h pixels high] 'Name' max. 9 characters.

<BUTTON name CREATE,"Label" STATIC|OWNERDRAW w,h>

Create button 'name' with Unicode "Label" (max. 55 chars.) w pixels wide, h pixels high].

<BUTTON name CREATE,"Text" CHECKBOX|GROUPBOX|RADIOBUTTON w,h>

Create check box, group box, or radio button 'name' with ANSI "Text" w pixels wide, h pixels high].

<BUTTON name FCOLOR|BCOLOR colorname>

Assign color to button.

<BUTTON name FLASH n>

Flash button n times.

<BUTTON name STATE ENABLE|DISABLE|SHOW|HIDE|CHECK>

Set state of hot button.

New Special Responses

See Pressing a Button.

See Also: Making Hotspots in Graphics

Variables

Variables can be used by the experienced courseware author to achieve a wide variety of effects. Many uses of variables in CALIScript replicate those found in general-purpose computer programming languages--as indexes in loops (numeric variables); as flags to show that certain conditions have been met (usually numeric variables); as algebraic symbols used in calculations (numeric variables); as containers for strings of characters which are acted upon repetitively (string variables); and so on.

One typical application of string variables, shown in the first sample exercise below, is in templates. You open the template in the Author Scratchpad and fill in the designated slots with specific information. Variables take care of the rest.

The second sample script shows how to use numeric variables as indexes in loops.

Another typical application of numeric variables, shown in the third sample script below, is in counting student responses.

Note: Variables may not be used inside CALIS commands, options, or special responses (enclosed within angle brackets). Thus it is unacceptable to substitute a variable for a parameter, as in the parameter "5" in the option <ALL: 5>

New Statement

% Variable Statement

As needed - variable declaration and initialization, of form %stringvariable1\$ = "value" or %numvariable = integer value.

Variable commands

<ASSIGN: a=b>

Assign the value b to the variable a.

<ASSIGN: n=n+1>

Increment variable n.

<SUBST:var\$>

Substitute value of variable.

A Sample Lesson Template Using String Variables

You can copy this topic to your clipboard, then paste it into your Author Scratchpad and try it! (Maximize your Help window before you copy, to make the lines as long as possible, since hard returns will be inserted by Windows wherever lines are wrapped, which will produce CALIS syntax errors in long script lines.)

C Template for "Name n X's", with \<ALL:n> and string variables.

C To use: 1) Fill in 'category' between quote marks ""

C 2) Fill in 'number' of required answers on next line

C 3) Fill in all correct answers between 'item' quote marks ""

- C 4) Enter the same number as in (2) on the #0 line of the script in
C the \<ALL:5> option, replacing the '5'.

@<MAXTRIES:99><GIVE:0>

%category\$ = ""
%number\$ = ""

%item1\$ = ""
%item2\$ = ""
%item3\$ = ""
%item4\$ = ""
%item5\$ = ""
%item6\$ = ""
%item7\$ = ""
%item8\$ = ""
%item9\$ = ""
%item10\$ = ""
%item11\$ = ""
%item12\$ = ""
%item13\$ = ""
%item14\$ = ""
%item15\$ = ""
%item16\$ = ""
%item17\$ = ""
%item18\$ = ""
%item19\$ = ""
%item20\$ = ""
%item21\$ = ""
%item22\$ = ""
%item23\$ = ""
%item24\$ = ""
%item25\$ = ""
%item26\$ = ""
%item27\$ = ""
%item28\$ = ""
%item29\$ = ""
%item30\$ = ""
%item31\$ = ""
%item32\$ = ""
%item33\$ = ""
%item34\$ = ""
%item35\$ = ""
%item36\$ = ""
%item37\$ = ""
%item38\$ = ""
%item39\$ = ""
%item40\$ = ""
%item41\$ = ""
%item42\$ = ""
%item43\$ = ""
%item44\$ = ""
%item45\$ = ""
%item46\$ = ""
%item47\$ = ""

```
%item48$ = ""
%item49$ = ""
%item50$ = ""
```

```
#0<ALL:5>
```

```
/<F2><TN Answers><SIZE 45,60,95,95>Answers already used:<POPUP F2 NOLOCK>
?<CUR:2><CE>Name at least <SUBST:number$> <SUBST:category$>.
?<CE>Enter your answers one at a time, each one followed
?<CE>by ENTER.
?<CE> <M:prompt>
?<CUR:prompt><CLEAR LINE>
```

```
+<IF:response$=item1$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item2$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item3$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item4$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item5$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item6$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item7$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item8$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item9$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item10$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item11$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item12$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item13$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item14$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item15$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item16$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item17$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item18$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item19$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item20$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item21$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item22$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item23$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item24$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item25$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item26$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item27$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item28$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item29$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item30$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item31$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item32$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item33$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item34$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item35$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item36$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item37$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item38$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item39$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item40$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item41$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item42$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item43$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item44$>;<F2><CUR:+1,2><LAST>
```

```
+<IF:response$=item45$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item46$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item47$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item48$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item49$>;<F2><CUR:+1,2><LAST>
+<IF:response$=item50$>;<F2><CUR:+1,2><LAST>
-<GIVEUP>;Correct answers:
&<SUBST:item1$>
&<SUBST:item2$>
&<SUBST:item3$>
&<SUBST:item4$>
&<SUBST:item5$>
&<SUBST:item6$>
&<SUBST:item7$>
&<SUBST:item8$>
&<SUBST:item9$>
&<SUBST:item10$>
&<SUBST:item11$>
&<SUBST:item12$>
&<SUBST:item13$>
&<SUBST:item14$>
&<SUBST:item15$>
&<SUBST:item16$>
&<SUBST:item17$>
&<SUBST:item18$>
&<SUBST:item19$>
&<SUBST:item20$>
&<SUBST:item21$>
&<SUBST:item22$>
&<SUBST:item23$>
&<SUBST:item24$>
&<SUBST:item25$>
&<SUBST:item26$>
&<SUBST:item27$>
&<SUBST:item28$>
&<SUBST:item29$>
&<SUBST:item30$>
&<SUBST:item31$>
&<SUBST:item32$>
&<SUBST:item33$>
&<SUBST:item34$>
&<SUBST:item35$>
&<SUBST:item36$>
&<SUBST:item37$>
&<SUBST:item38$>
&<SUBST:item39$>
&<SUBST:item40$>
&<SUBST:item41$>
&<SUBST:item42$>
&<SUBST:item43$>
&<SUBST:item44$>
&<SUBST:item45$>
&<SUBST:item46$>
&<SUBST:item47$>
&<SUBST:item48$>
```

```
&<SUBST:item49$>  
&<SUBST:item50$>  
-*,<CE>Check your spelling.  
-&;Please type one word only!
```

```
#end
```

<WHILE: variable LT value>

<WHILE: variable = value>

Executes a loop while variable is less than [equals] value.

Example:

```
%counter=0  
%lines=10  
%line$="This is line "  
#0  
?This is an example of while command  
?<WHILE: counter < 10><CE><ASSIGN:  
counter=counter+1><SUBST:line$><SUBST:counter>.<CUR: +1,0>  
+Correct  
#end
```

<IF:variable=value>

<IF:variable GT value>

Evaluates the expression variable = [GT, LT] value and executes the preceding command or special response if true.

Special Responses

<IF:response\$=variable\$>

Defines action by WinCALIS if the student's text response, stored in response\$, matches the value of variable\$,

<IF:mark\$=variable\$>

or text marked, stored in mark\$, matches the value of variable\$,

<IF:numvariable=value>

or expression numvariable=value is true.

A Sample Lesson Using Variables as Counters

In the sample lesson below there is one response if the student answers correctly on the first try, and another response if it is his second or greater try.

```
#0<NOSC><NOPR>
```

```
?<ADD: 1--4>
#end
```

```
!Give the antonym for each word.
!Words are presented in antonym pairs, to make them easy to memorize quickly.
!
!If you enter the correct antonym on the first try, it is assumed that you know
!the pair already, and you will not be asked to type the other half of the pair.
```

```
%index=0
```

```
#1
?<ASSIGN:index=index+1> 1. sweet
+sour<IF:index=1>;Correct<DEL:2><ASSIGN: index = 0>
+sour;<CE>Correct<ASSIGN: index = 0>
```

```
#2
?<ASSIGN:index=index + 1> 2. sour
+sweet<IF:index = 1>;Correct<DEL 1><ASSIGN: index = 0>
+sweet;Correct<ASSIGN: index = 0>
```

```
#3
?<ASSIGN:index = index + 1> 3. cold
+hot<IF:index=1>;Correct<DEL 4><ASSIGN: index = 0>
+hot;Correct<ASSIGN: index = 0>
```

```
#4
?<ASSIGN:index=index+1> 4. hot
+cold<IF:index=1>;Correct<DEL 3><ASSIGN: index = 0>
+cold;Correct<ASSIGN: index = 0>
```

Global Variables

mark\$

Stores text marked with mouse by student.

Sample of mark\$ variable:

```
#0
?Please mark any portion of this text the using mouse and then click on the button.
?You get special messages if you mark the words "portion" or "button".
?<CE><BUTTON: mark, CREATE, "Mark">
$<BUTTON: mark><IF: mark$ = "portion">;Great! you are doing a good job, keep on going.
$<BUTTON: mark><IF: mark$ = "button">;That's easy, isn't it?
$<BUTTON: mark>;<F1>"<LASTMARK>"<EXPORT: NEW temp.txt>
$<BUTTON: mark>;<F1>"<LASTMARK>"<EXPORT: APPEND temp.txt>
+Correct
#end
```

response\$

Stores text of last student response (Same as <LAST> command).

result

Stores result (0 or 1) of certain commands, such as <DONE>.

Task Groups

New Task Options

<GROUP:grouptask [,exitgrouptask]>

Identifies group tasks to be executed upon entry to and exit from group.

<GROUPLOCK>

Prevents the student from passing out of the current group until all tasks in the group are completed.

For further details see Group Behavior.

Keeping Student Records

New Commands

<DONE: tasklabel>

Determine whether task 'tasklabel' has been done and store result in global variable "result."

<EXPORT NEW|APPEND filename>

Export contents of current window to a file.

Sample of <EXPORT> command:

```
#0
?Please mark any portion of this text the using mouse and then click on the button.
?You get special messages if you mark the words "portion" or "button".
?<CE><BUTTON: mark, CREATE, "Mark">
$<BUTTON: mark><IF: mark$ = "portion">;Great! you are doing a good job, keep on going.
$<BUTTON: mark><IF: mark$ = "button">;That's easy, isn't it?
$<BUTTON: mark>;<F1>"<LASTMARK>"<EXPORT: NEW temp.txt>
;$<BUTTON: mark>;<F1>"<LASTMARK>"<EXPORT: APPEND temp.txt>
+Correct
#end
```

<REPORT>

Display a detailed task-by-task score report.

<PRINT [W]>

Send to the default printer the contents of the current window or the window specified.

New Option

<QUIZ>

Turn on “Quiz” mode, require student to log in and record student responses in disk file 'nameSSno.scr'.

See Also: Turning an Exercise into a Quickie Quiz.

More New Features of CALIScript

New Statement

L Default font

One per script, font name enclosed in double quotes (e.g., L"Arial") - may occur anywhere, but will take effect at start.

New Commands

<CLOSE: window>

Close DIR or tutorial window Fn opened with <POPUP [W] NOLOCK> command.

#0

?<BLA><CE>Click on the 'Close' button to close the tutorial window.

?<F1><BLA><CE><SIZE: 25, 25, 50, 50>Please wait...

?<Q><CUR:13><CE><BUTTON: nags, CREATE, "Close", 80, 25>

?<CUR:5><CE> <M:prompt><CUR: prompt><POPUP F1: NOLOCK>

\$<BUTTON: nags>;<CLOSE: F1>

+Great;

#end

<COMMENT>

Ignores the rest of the current line in generating *.WCL file.

For example:

#0

?This shows the comment command.

?Start comment <M: prompt><COMMENT>This text will not go into the *.WCL file

?<COMMENT>The Author program can make use of this command to comment (out) the portion of the CALIS script that

?<COMMENT>explains, to the teacher or instructor, a particular command or topic that it automatically generates.

?<CUR:prompt>

+correct

#end

<DELAY: OFF>

Terminates any previously issued DELAY command.

Sample of <DELAY: OFF>:

#0

?<BLA><CE>Click on the 'Delay' button to come out of the Delay loop.

?<CUR:3><CE><BUTTON: nags, CREATE, "Delay", 80, 25>

?<CUR:7><CE> <M:prompt><CUR: prompt><DELAY: 5>

\$<BUTTON: nags>;<DELAY: OFF>

+Great;

#1

?<BLA><CE>Click on the 'Stop' button to stop the timer as well as to close the tutorial window.

?<F1><BLA><CE><SIZE: 25, 25, 50, 50>Please wait...

?<Q><CUR:13><CE><BUTTON: nags, CREATE, "Stop", 80, 25>

?<CUR:5><CE> <M:prompt><CUR: prompt><POPUP F1: NOLOCK><DELAY: 5><CLOSE: F1>

:\$<BUTTON: nags>;<DELAY: OFF><CLOSE: F1>

:\$<KEY: ANY>;<DELAY: OFF><CLOSE: F1>

\$ok;<DELAY: OFF><CLOSE: F1>

+Great;

#end

<DRAW: object_name, object_type [, width, height, border_thickness, border_color, fill_color]>

no abbreviation

Parameters:

object_name: Unique label for object drawn

object_type: BOX or LINE

width: Width of the object in pixels (x increment from the current position).

height: Height of the object in pixels (y increment from the current position).

border_thickness: Thickness of the border in pixels.

border_color: Color of the border pixels.

fill_color: Inside color of box.

Draws an object.

Currently supported object types are BOX for a box and LINE for a straight line (which may be horizontal, vertical or sloping). Future object types might include ARROW, DOTTED_LINE, DOTTED_BOX, etc.

For example:

? <DRAW: menubox, BOX, 320,330,3,BLUE,YELLOW>

?

? This text will appear

? inside the box.

The <DRAW> command in the above example will draw a box 320 pixels wide and 330 pixels high, with a blue border 3 pixels in thickness and a yellow box background. The text will appear inside the box.

<KEY ANY>

Defines action taken immediately by WinCALIS if any key is pressed.

<KEY controlkey>

Defines action taken immediately by WinCALIS if one of the following control keys is pressed: SHIFT, CONTROL, ALT, BACK, TAB, CAPITAL, ESCAPE, SPACE, PRIOR, NEXT, END, HOME, LEFT, UP, RIGHT, DOWN, DELETE, NUMPAD0, NUMPAD1, NUMPAD2, NUMPAD3, NUMPAD4, NUMPAD5, NUMPAD6, NUMPAD7, NUMPAD8, NUMPAD9, MULTIPLY, ADD, SEPARATOR, SUBTRACT, DECIMAL, DIVIDE, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, NUMLOCK.

<LASTMARK>

Prints to the screen the last text marked by the student.

Sample script using <LASTMARK>:

```
#0
?Please mark any portion of this text using the mouse and then click on the button.
?You get special messages if you mark the words "portion" or "button".
?<CE><BUTTON: mark, CREATE, "Mark">
$<BUTTON: mark><IF: mark$ = "portion">;Great! you are doing a good job, keep on going.
$<BUTTON: mark><IF: mark$ = "button">;That's easy, isn't it?
$<BUTTON: mark>;<F1>"<LASTMARK>"<EXPORT: NEW temp.txt>
;$<BUTTON: mark>;<F1>"<LASTMARK>"<EXPORT: APPEND temp.txt>
+Correct
#end
```

<MCICTRL: SHOW|HIDE>

Show or hide MCI Control Bar (Media Controls Bar) -- student controls multimedia.

<PASS [n]>

Pass forward to next task or forward n tasks.

<PASSBACK [n]>

Pass back to previous task or back n tasks.

<SET:[W] ATTRIBUTE BOLD>

Sets the window font to bold {<SET:[W] ATTR BO>}.

<SET:[W] ATTRIBUTE ITALIC>

Sets the window font to *italics* {<SET:[W] ATTR IT>}.

<SET:[W] ATTRIBUTE STRIKEOUT>

Sets the window font to ~~strikeout~~ {<SET:[W] ATTR SO>}.

<SET:[W] ATTRIBUTE UNDERLINE>

Set window font to underline {<SET:[W] ATTR UL>}.

<SET:[W] ATTRIBUTE [NORMAL]>

Sets the window font to normal {<SET:[W] ATTR [NO]>}.

Sample script showing <SET ATTRIBUTE> commands:

```
C<SET ATTR IT>This text should be in italics.
CAnd this should still be...
@<MAXTRIES:99><GIVE:1>
#0<NOSC><NOPR>
!<CE><SET DIR ATTR UL><SET Q ATTR SO>
!<SET D ATTR BO><COMMENT>This command has no effect, since the
!<COMMENT>D window does not yet exist.
!<CE>This is a demo of various window attributes. This is underlined.
!<F3><CE><SET ATTR IT>This is all in italics
!<CE>all through here and at this point <SET ATTR> goes back to normal!
?<ADD: 1--1>
?<ADD: 2--2>
#end

#group1<NOSC><REP>
!<BLA><CE>This text is underlined.
!<CE>And this should still be...

#1<GROUP:group1>
/<BLA><SET ATTR BO><SET ATTR UL>This text is bold AND underlined.
/And this should still be...
?<BLA><CUR:1><CE>This text is to be deleted and should be struck out.
?<CE>And this should still be...
?<CE>How about the F3 window?
?<CE><M:t1><UL>T<UL>rue<M:t2>      <M:ff1><UL>F<UL>alse<M:ff2>
+<MOUSE:t1 t2> <OR> <KEY: 't'>;<CE>Not bold, right?
-<MOUSE:ff1 ff2> <OR> <KEY: 'f'>;<CE>Not bold, right?
-&;Please type the letter of one of the choices listed above.

#group2<NOSC><REP>
!<BLA><CE><SET ATTR NO>Back to Normal.

#2<GROUP:group2>
/<BLA><SET ATTR NO>Presentation (Text) Window back to Normal.
```

```
/Recheck F3...<F3><CE>Here's some new text in the F3 window...<T>
?<BLA><CUR:1><CE><SET ATTR NO>All attributes should now be reset to normal.
?<CE><F3>
?<CE><SET ATTR IT><SET ATTR UL>Further text for F3 window, now underlined italics.
?<CE><Q>
?<CE><M:t1><UL>T<UL>rue<M:t2>      <M:ff1><UL>F<UL>alse<M:ff2>
+<MOUSE:t1 t2> <OR> <KEY: 't'>;<CE><SET D ATTR BO>This should now be bold, since
&<CE>the SET BOLD command was inserted after this D window was created.
&<SET ATTR UL>Underlined?
&And this should still be...
&<SET ATTR NO>Back to Normal?
&And this should still be...
-<MOUSE:ff1 ff2> <OR> <KEY: 'f'>;<CE><CE><SET D ATTR BO>This should now be bold, since
&<CE>the SET BOLD command was inserted after this D window was created.
&And this should still be...
&<SET ATTR UL>Underlined?
&And this should still be...
&<SET ATTR NO>Back to Normal?
&And this should still be...
-&;Please type the letter of one of the choices listed above.
```

<SET:LANG langname>

Use “langname.ini” to set pixel heights for standard font sizes for subsequent fonts.

Even More New Features in WinCALIS:

WinCALIS and WinCALIS Author:

- 1) Enhancements to Help menu virtual (on-screen) keyboard:
 - while typing, pressing Shift or Ctrl shows the shifted state or state with Ctrl pressed of the virtual keyboard
 - characters can be inserted in text directly from virtual keyboard with left mouse click
 - the effect of overstrike accents on other (accented) characters is shown with a right mouse click, and the displayed accented characters with left mouse click
 - clearer keycap characters
 - more legible placement on keycaps of stacked characters such as in Thai
- 2) Both programs now each support passing forward and backward from task to task with either Ctrl+N and Ctrl+P or Gray + and Gray -
- 3) Unicode Browser in WinCALIS Options menu and Unicode Input in Author Edit menu have been completely updated to Unicode 2.0 and beyond, including placement for draft proposals for language/scripts already approved by the Unicode Technical Committee

WinCALIS:

- 4) Options menu now has a "Previous Answer" option, to remind user of "doskey" functionality of Up Arrow key
- 5) Ability to hide WinCALIS title bar and menus
with following line in [WinCALIS] section of wincalis.ini file:
Menus=No
(default is "Menus=Yes")
- 6) Ability to run WinCALIS in a "lowest-common-denominator"
640x480 VGA window with following line in [WinCALIS] section of
wincalis.ini file:
LessonWindow=VGA
(the default) or run maximized with line
Window=MAX
- 7) Enhanced default score report to user
- 8) Enhancements to formatting of score results with <REPORT> command
- 9) Audio recording by the user now supports recording at 11025, 22050,
or 44100 samples per second and 8 or 16 bits per sample. Recording
rate is set with the following lines in the [WinCALIS] section of the
wincalis.ini file:
WaveSamplesPerSec=11025
WaveBitsPerSample=8
Using the higher settings, WinCALIS has been successfully used as a
tool for automated audio data collection.

WinCALIS Author:

- 10) Format menu enhancements, including insertion of various
formatting commands by clicking on menu items
- 11) Semi-automated marking of "hotspot" hypertext from Format
menu "Insert Mark" item
- 12) Menu-driven insertion of CALIScript commands, special responses,
options, variables, etc. in Tools Menu and Short Answer dialog box
- 13) Edit menu on-the-fly code conversions with "Paste As..."
- 14) More code conversion options
- 15) Improved user message warnings on exiting without saving

Chapter 9: Specific Language Reference Manuals

Miscellaneous Language-Related Hints

Formatting Multiple Choice and True/False Questions

For both Multiple Choice and True/False task types, WinCALIS Author uses the first letter in the Anticipated Response text box to generate a special keyboard response, making it possible for the student to respond by pressing the key corresponding to that letter without pressing ENTER. Author permits you to type any valid Unicode character in that box, but the first character should preferably be one which appears on the keyboard, to avoid unexpected results. Be creative. In some languages, “O” symbolizes “true,” while “X” symbolizes “false.” Thus “O” and “X” can be used as the first letter in each line, followed by a space and then a word in the appropriate language.“

Fixed Width Blanks in Cloze Exercises

In certain language keyboards, such as the romanization-conversion keyboards in Chinese, Japanese, and Korean, what you type at the cursor is sometimes longer than it is after conversion has taken place. For example, Japanese *watakushi* (11 keystrokes, 4 *kana* characters) converts to 1 *kanji* character; Korean *ha* (2 keystrokes) converts to 1 *hangul* character; and Chinese *shuangsheng* (11 characters) converts to 2 *hanzi* characters. In order to permit the typing of the longer romanized string, before conversion takes place, the length of the blanks in a cloze exercise should always be set sufficiently long to permit the typing of romanized input, not the final answer the student will enter after conversion has taken place.

However, the WinCALIS Author default setting is to make each blank exactly as long as the longest Correct/Continue or Incorrect/Remain Here answer. And WinCALIS Author always sets the WinCALIS <LEN> option so as to prevent the student from typing more than the space permits, once the maximum number of characters is reached, since this would overwrite the contextual text. You should always change variable blank length to fixed blank length. You can select fixed blank length and set the actual length to be used in the Cloze Options dialog box, available under the Script menu.

Be sure that you test all your Correct/Continue and Incorrect/Remain Here answers to ensure that they can actually be entered by the student. You will be able to determine what is needed through a little experience. Usually it is enough to add a few extra characters to the number calculated by Author to be the longest answer. But in the case of *watakushi* above, in order to be able to enter the single *kanji* character for *watakushi*, the student will need to have space to type wa (2->1) ta (2->1) ku (2-1> shi (3->1) for a total of 6 characters (1+1+1+3). If all your blanks were a single character, you would need to set them all to at least 6 characters long.

Default Font Attributes for Answers in Cloze Exercises

The default font attributes for the Correct/Continue answer shown in the blank in cloze exercises is “bold” and “blue.” This is acceptable for Roman script and perhaps other scripts, but “bold” does not look good for a number of scripts, including Han characters, and the default blue color maynot be appropriate for all cultural contexts. These defaults can easily be changed in the Cloze Options dialog box, available under the Script menu.

Arabic and Hebrew (Right-to-Left) Scripts Reference

Right-to-Left and Left-to-Right Languages Basic Information

Cursor Position

The position of the cursor at the beginning of a line is determined by the direction of the chosen language. For example, if you choose Arabic the cursor will appear on the right side of the window. If before typing, you switch to English the cursor will flip to the left of the window.

Cursor Movement

The direction the cursor moves is determined by the first non-neutral/directional character of the line. If the first character encountered by WinCALIS editor is of type left to right, the cursor will follow that direction until the line break (you press ENTER) even if the language type has changed in the middle of the line. The same holds true if the first character WinCALIS editor encounters is of the form right to left. In those instances in which a neutral character begins a line WinCALIS will not make a judgment until it encounters the first non-neutral character.

Text Order

Although the directional movement of the cursor does not change within a line, all text will appear in its proper order. For example, if the cursor is moving from left to right and you change to a language written from right to left, the text will be 'pushed' out from the cursor point. If you switch languages again the cursor will jump over the text it 'pushed' out and continue the line. This will occur in the same manner if the cursor is moving right to left

Switching Directions

When mixing right-to-left text and left-to-right text, a maximum of 13 changes of direction is permitted per line (between hard returns). This limit can easily be exceeded if numerous CALIS commands are used in text, since CALIS commands are left-to-right in directionality. A typical problem case is that of marking text, when "marks" such as "<M:a1>" are inserted in right-to-left text to identify the beginning and ending of "hotspot" hyperlinks. A rule-of-thumb to go by is that a maximum of three marked words or phrases may be used successfully in a single line--four, if one of the marks is at the beginning of a line (which will not require a change of direction, since the CALIS statement marker at the beginning of the line is already left-to-right). Usually, line lengths can be edited so as to avoid inconvenience.

See Also: Miscellaneous Language-Related Hints

Chinese 語文教學 · 语文教学

There are three different Chinese input methods available:

- Pinyin Phonetic Conversion
- Cangjie Input Based on Character Shape
- Character Input through Browsing Unicode

Of these, the normal method for typing and editing text is the first, pinyin phonetic method, which has many sophisticated features to enhance efficiency. The other methods are intended to supplement pinyin input, when you need to type a rare character, or a character the pronunciation of which you are uncertain of.

There is also a keyboard available for typing the Chinese Phonetic Alphabet (“Zhuyin Fuhao” or “Bopomofo”) characters. This is not a method for entering Chinese characters.

The Pinyin Phonetic Input Method

This technique is a phonetic conversion method, based on entering Modern Standard Chinese in phonetic transcription in the Pinyin romanization, with or without tones. WinCALIS Author editor takes the pinyin text and converts it into the accurate Chinese characters each time you press the space bar or another conversion function key after typing the phonetic version of a Chinese word or character.

Note: When mixing Chinese with any Roman-script language (such as English), a space, parenthesis, or other punctuation mark should separate the Roman language from the pinyin in order to obtain a proper conversion.

Conversion Preference

WinCALIS Author editor takes any unconverted pinyin to the left of the cursor and converts it when the space bar is pressed. If the string of characters to the left of the cursor does not constitute a valid pinyin syllable or syllables, a message “Invalid Pinyin” will appear on the screen.

WinCALIS Author handles the conversion process through a lookup table connected to an internal dictionary of Pinyin-Hanzi equivalents, with over 40,000 entries, including about 17,000 Chinese characters in both simple (modern) and complex (traditional) styles, as well as polysyllabic words. If the correct character(s) cannot be determined, because there are several homophones with the same pinyin, a Chinese Selection dialog box will appear and display some homophone alternatives, arranged in order of descending frequency.

To Convert or Not to Convert?

In order to convert a collection of pinyin words previously typed, place the cursor at the end of the pinyin phrase to be converted and press the space bar or other appropriate function key. If a match can not be found within the dictionary, a message “Invalid pinyin” is displayed. Either press the space bar or use the mouse and click on the OK button to clear the message. You can move the cursor back one syllable at a time, until part of the phrase is converted, then convert the remaining phrase or choose individual syllables and convert them one at a time with the [F7] or [F8] function key.

Watch Your Tones!

When typing Chinese in WinCALIS, the words must not contain any spaces or hyphens in pinyin transcription. Enter tone marks with the number keys 1-4, then press the most suitable of several different Convert keys. The internal dictionary will search for a correct character translation and try to match what you have typed. A Chinese translation is either inserted at the cursor position of your text, or if more than one possible word is detected the Chinese Selection dialog box will appear and you can

choose the appropriate homophone. To indicate your selection click the mouse on the word of your choice, or press the space bar to make the Selection Window active, move up and down the list with the arrow keys until the desired word is highlighted, then press ENTER. You can cancel the window without making a selection by pressing the ESC key.

What is Your Pinyin Status?

After each conversion, the correct toned pinyin is displayed in the Chinese Status Bar at the bottom of the screen. The display of the pinyin text is intended to help you become familiar with the entry that was successfully converted by WinCALIS, so that you may make an informed choice in situations where there is some uncertainty as to the correct pinyin form, tone, and so on. This is particularly handy when you type a pinyin form without tones and press the F8 key for an all-tone search of the dictionary. By typing a toned form you can usually avoid the need to select from the homophone selection window.

You have a number of options for converting pinyin to characters. Most of the options are selected with either the space bar or one of the function keys.

Functions of the Space Bar

When Chinese is the selected language in WinCALIS or WinCALIS Author, the space bar functions as follows:

- **Chinese Space**

When there is no pinyin text preceding the cursor, the space bar inserts a blank space, the length of one Chinese character, which is equal to over two English or pinyin-size characters, at the current cursor position.

- **Convert-Word Key**

When the space bar is pressed after typing a word in pinyin transcription including the proper tones, WinCALIS will search its internal dictionary for a free-form word which fully matches the typed pinyin pronunciation, including tone: e.g., , , etc. All characters which are only bound forms (parts of words, e.g.) are excluded, with the exception of the special case of capitalized proper nouns. If more than one word with the same pronunciation (homophone) is found, the choices are displayed in the Chinese Selection dialog box. If you do not see the desired word in the Chinese Selection window, particularly in the case of single characters, cancel the window by pressing the ESC key, then press function key F7 for a full character search. Or simply press F8 for an all-tone search (tone is ignored).

If a proper noun, such as a surname (e.g., Zhèng), the name of a dynasty (e.g., Táng), or a place name (e.g., Beijing), is typed with an initial capital letter, it is converted directly when the space bar is pressed, even if it is a bound form or there are other non-proper-noun homophones. If there are two or more such matches (e.g. Zhang, Zhang), they are displayed in a selection window.

Selecting Options from the Chinese Selection Dialog Box

When a Chinese Selection box is displayed in the screen, a word can be selected by using the arrow keys to move from the first initial highlighted pinyin to the desired word as an alternative to using the mouse.

The vowel ü (u-dieresis) (u with two dots over it) is typed with the ‘v’ key or CTRL + u. The rare vowel ê (e-circumflex) is typed with CTRL-e. Each of these vowels is also modifiable by the tone marks 1-4.

When tone marks are added to a syllable, they are automatically placed over the correct vowel in each syllable, as long as they are typed after the first occurring vowel in each syllable. By convention, students should be encouraged to enter the tone right after the vowel it will appear over (like dotting an “i” or crossing a “t” immediately in printing) or at the end of each syllable (like dotting “i”s and crossing “t”s after the end of a word in handwriting).

Certain phrase-final punctuation marks also serve as convert keys during Chinese input, thus making it unnecessary to first press the space bar, then press the desired punctuation mark key. (Not yet implemented.) These marks include the period (。), clause comma (，), serial comma (、), exclamation

point (。), question mark (?), and “yuandian” (圆点). Each of these marks is typed by pressing the designated key on the keyboard, which is also used for the smaller corresponding English punctuation marks during English input. This is also true of the Chinese-spaced quotation mark and Chinese space. Other punctuation marks unique to Chinese are entered by pressing special keys. These are as follows:

Chinese serial comma (、)	Grave accent key (`)
Angle-bracket open quote (「)	Shift+2
Angle-bracket close quote (」)	Shift+3
Double angle-bracket open quote (『)	Shift+4
Double angle-bracket close quote (』)	Shift+5
"Yuandian" 圆点 separator mark (•)	Shift+6
Chinese ditto mark (々)	Shift+7

All keyboard key assignments can be displayed by clicking on main menu Help then selecting menu item Keyboard Help. A virtual keyboard picture will come up on the screen.

Chinese Function Keys

You can use the function keys on your keyboard to perform many tasks faster:

F1	Get help on the selected command, dialog box, or message
F2	Highlight the character to be accented (Accent Overstrike Key)
F3	Switch between two currently used languages
F4	Reselect (“Oops”) key. (WinCALIS) If a word is incorrectly converted, press F4 immediately to display the Selection Window dialog box. Reselecting the proper word from the Selection Window will automatically replace the incorrectly converted word. Save current script file and continue editing (WinCALIS Author)
F5	Look up highlighted word in On-line Dictionary
F6	Expand Scope of Selection (Selection Window) Broadens the range of choice in the Chinese Selection Window when pressed during display of a Chinese Selection window. Allows student to give up (WinCALIS)
F7	Character Search Initiates a dictionary search for all entries in the dictionary which match the typed pinyin, including tone, regardless of whether they are free words or bound morphemes.
F8	All-tone Dictionary Search Initiates a dictionary search for all characters or words which match typed pinyin, regardless of tone; any tone marks displayed will be ignored. In All-Tone mode, initiates a search for only those characters or words with the tone typed.
F9	Reselect (“Oops”) key. (WinCALIS Author and UniEdit) If a word is incorrectly converted, press F9 immediately to display the Selection Window dialog box. Reselecting the proper word from the Selection Window will automatically replace the incorrectly converted word.
F10	Default Selection/Duplicates the last word entered. Avoids selection from the Chinese Selection window by automatically selecting and displaying either the most common among two or more homophone words or the one most recently selected from a Chinese Selection window (ignores bound forms); each time a different selection is made, that becomes the new “default” value. If pressed without any pinyin at the cursor, F10

	repeats the last selection made from a selection window (useful for reduplication, or for quick conversions of the same word in multiple locations).
F11	Play Audio (WinCALIS)
F12	Play Video (WinCALIS)
Alt+F3	Set Cache/No-Cache Mode In Cache mode, entries selected from the main dictionary on disk are added to a temporary cache, and displayed even in No-Disk mode along with memory-dictionary entries. No-Cache mode disables this feature.
Alt+F4	Close the application
Alt+F5	Set Convert/No-Convert (literal Pinyin) In Convert mode, pinyin text at the cursor is converted into Chinese characters. In No-Convert mode, pinyin text is left unconverted.
Alt+F7	Alternate between Word and Character Mode. In Word Mode, pressing the space bar searches for words (free forms) only. In Character Mode, pressing the space bar searches for all forms, both free and bound.
Alt+F8	Alternate between using the space bar for Tone Mode and All-Tone Mode. In Tone Mode, pressing the space bar searches for toned forms only. In All-Tone Mode, pressing the space bar searches for all forms, regardless of tone.
Alt+F9	Alternate between Simple and Complex character style. Changes the conversion of the space bar and other encode keys between complex-style (traditional) and simple-style (modern) characters.
Alt+F10	Alternate between Default and No-Default Search. In Default Mode pressing the Space Bar converts the default form either the most frequently occurring form or the most recently selected form) without any Selection Window. In No-Default Mode pressing the space bar displays a Selection Window.
Alt+F11	Displays the Chinese Dictionary Search Option Dialog Box.
Alt+F12	Alternate between Disk Mode and No-Disk Mode. In Disk Mode, pressing the space bar immediately searches both the memory and disk dictionaries, producing a larger selection window, with more selections. In No-Disk Mode, pressing the space bar searches the memory dictionary and the cache for matching entries. Single entries are directly converted. If there is a Selection Window, further entries from the main disk dictionary are displayed by pressing the "More" button or F6.

Your favorite settings may be saved as part of the WinCALIS "desktop" in WinCALIS Author, by clicking on Script, Desktop, Save. The original settings are "Word, Tone, No-Default, Cache, Convert, Simple," which for the typical author or student user should produce the most efficient and accurate encoding of pinyin phonetic input into Chinese characters.

Chinese Status Bar

When Pinyin-to-Hanzi is the current language keyboard, a Status Bar at the lower left corner of the screen displays the current settings for Chinese input.

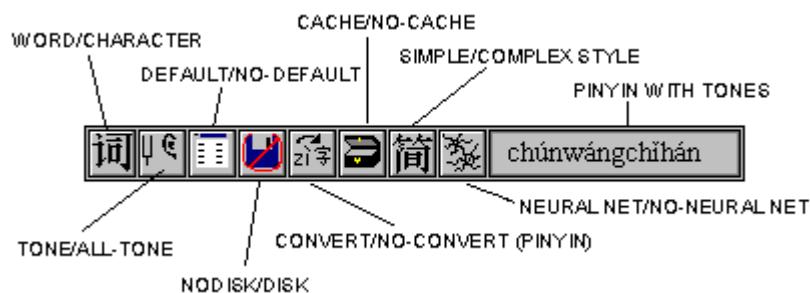


Figure 34: Chinese Status Bar

These settings can be changed by clicking on the buttons with the mouse, by pressing the corresponding Alt + function key combination, or by displaying the Chinese Dictionary Search options dialog box with Alt+F11 and clicking on the desired check box. You can view the following in the Status Bar:

the WORD/CHARACTER button,
the TONE/ALL-TONE button,
the DEFAULT/NO-DEFAULT button,
the NODISK/DISK button,
the CONVERT/NO-CONVERT (Pinyin) button,
the CACHE/NO-CACHE button,
the SIMPLE STYLE/COMPLEX STYLE button,
the NEURAL NETWORK/NO-NEURAL NETWORK button and
the Pinyin word, showing the correct toned form of the last word converted from Pinyin to Hanzi.

Special Pinyin Forms

Certain words occur so often that in addition to their regular forms, they have special alternative toneless or non-standard pinyin forms which make them easy to enter, without the need for selection from the Homophone Selection Window. Here is a list of some of these ad hoc forms. Note: This list includes just a sample of the neutral-tone “particles” which are normally typed without tone, such as “le,” “ne,” “zhe,” etc.

不	b, bu (and bú, bù)	她	taa (and tā)
的	d (and de)	他们 (他們)	tamen (and tāmen)
地	di (and de, dì)	她们 (她們)	taamen (and tāmen)
个 (個)	ge (and gè)	它	tuó (and tā)
过 (過)	guo (and guò)	也	ye (and yě)
很	hen (and hěn)	一	yī (and yí)
就	jiu (and jiù)	在	zai (and zài)
里 (裏)	li (and lǐ)	在	zay (and zài)
儿 (兒)	r (and ér)	再	zhe (and zháo, zhǎo, zhuó, zhù)
是	shi (and shì)	着 (著)	zi (and zǐ)
他	ta (and tā)	子	

National Phonetic Alphabet (“Zhuyin Fuhao” or Bopomofo)

Under the Chinese language submenu, selection of the “Bopomofo” language enables entry of the National Phonetic Alphabet symbols widely used in Taiwan, and commonly known as bo-po-mo-fo, in your text. The keyboard mapping is the standard one used in Taiwan. Note that this is not a bo-po-mo-fo input method, but rather a way of entering these symbols in your text, such as might be desired in a glossary or annotated text.

Zhuyin Fuhao-Pinyin Correspondence Table

For users familiar with the Chinese Phonetic Alphabet (“Zhuyin Fuhao” or “Bopomofo”), but not with Pinyin romanization, the following chart of rough equivalents may be useful:

ㄅ b ㄆ f ㄇ g ㄏ h ㄓ zh(i) ㄔ ch(i) ㄕ c(i) ㄖ r(i) ㄗ z(i) ㄘ x ㄙ s(i) ㄥ e
 ㄉ d ㄊ t ㄋ k ㄌ l ㄍ g ㄎ h ㄏ h ㄓ zh(i) ㄔ ch(i) ㄕ c(i) ㄖ r(i) ㄗ z(i) ㄘ x ㄙ s(i) ㄥ e
 ㄐ p ㄑ q ㄒ x ㄓ zh(i) ㄔ ch(i) ㄕ c(i) ㄖ r(i) ㄗ z(i) ㄘ x ㄙ s(i) ㄥ e
 ㄓ zh(i) ㄔ ch(i) ㄕ c(i) ㄖ r(i) ㄗ z(i) ㄘ x ㄙ s(i) ㄥ e
 ㄗ z(i) ㄘ x ㄙ s(i) ㄥ e
 ㄘ x ㄙ s(i) ㄥ e
 ㄙ s(i) ㄥ e
 ㄥ e

The Cangjie Input Method

Summary of Main Principles

The Cangjie method of computer input, which is also known as the Dragon method, is a fast, reliable alphabetic typing method for Chinese characters, based on their shape, without regard to their pronunciation. It was named by the inventor of the method, Chu Pang-fu, in Taiwan, after the legendary inventor of writing in Chinese antiquity, Cangjie (in Japanese, pronounced Sôketsu).

We have created for WinCALIS and WinCALIS Author an extended, more user-friendly version of the original Cangjie method, which makes better use of the Windows graphical interface and mouse. You can select this form of character shape-based entry under the L:[Language] menu, by selecting the “Cangjie (by shape)” keyboard in the Chinese submenu, the “Kanji by shape” keyboard in the Japanese submenu, or the “Hanja by shape” keyboard in the Korean submenu.

When you select one of these keyboards, a dialog box appears. You can enter the letter-symbols representing component shapes either by typing them from the keyboard in the text box or by clicking on the piano-keyboard-style buttons, which show the letters and their corresponding shapes.

Here is a description of the essentials of the original Cangjie method:

The Cangjie method allows use of the ordinary computer keyboard to encode Chinese characters, by assigning 24 of the 26 letters of the Roman alphabet to represent the most common component elements of Chinese characters, grouped into 24 archetypal categories, each with a full-character label. Every Chinese character has a unique Cangjie code of from 1 to 5 letters. A space typed after a sequence of letters signals the computer system that the code is complete and initiates a lookup routine. (No space is required after a five-letter code.)

The letter “X” is a special symbol used in a small number of cases to disambiguate two characters which would have the same Cangjie input code. The letter “Z” is used in some versions of the Cangjie

method to signal a user-defined character or phrase, or at the end of a Cangjie, as a disambiguator, like an “X” at the beginning. In WinCALIS and UniEdit an “X” at the beginning was used uniformly, but now “Z” at the end is generally used instead, since this fits the new user-friendly interface better.)

Mnemonic symbols

The 24 letter-symbols belong to four broad mnemonic categories:

- traditional Chinese natural philosophy (the seven heavenly luminaries and their correlates in the five “elements”)

(A) (“sun”) (B) (“moon”) (C) (“metal”) (D) (“wood”)
(E) (“water”) (F) (“fire”) (G) (“earth”)

- character stroke-types

(H) (“left-downward sloping stroke”) (I) (“dot”) (J) (“cross”)
(K) (“criss-cross”) (L) (“vertical”) (M) (“horizontal”)
(N) (“hook”)

- the human body

(O) (“man”) (P) (“heart/mind”) (Q) (“hand”) (R) (“mouth”)

- typical character shapes

(S) (“box open on the side”) (T) (“pair”)
(U) (“box open on top”) (V) (“angle ‘V’ shape”)
(W) (“large box”) (Y) (“dot plus line”)

As may be seen from some of the above examples, several of the letters also have a mnemonic association through their own shapes:

- “V” represents V-shaped components
- components looking like half a square “S” are represented by “S”
- U-shaped components are represented by “U”
- a small “l” is an example of a vertical stroke, represented by “L”
- a “dot” is suggested by the “dot” of the lower-case “i”
- the hook or angle of the “N” represents hook-shaped components

Order of Encoding

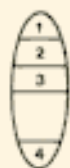
In entering the Cangjie code, the order of letters is similar to, but not completely identical with, the traditional stroke-order of Chinese characters. A character is built up left-to-right, top-to-bottom, and from outside to inside. When a character is so complex that it has more than the necessary number of components to uniquely identify it, the left-hand side and the first and last components of the right-hand side are usually encoded, with the middle of the right-hand side omitted.

In a typical character with a “radical” component on the left side, the radical is encoded with either one letter for the entire radical, or two letters, representing its first (top) and last (bottom) component parts. The remaining part of the character is encoded with the minimum number of letters to distinguish it from any other characters.

Chart Showing Cangjie Composition Principles

取碼規則：取碼順序為從上至下，從左至右，從外至內

整體字取碼：



商
YCBR

字身可分為次字首次字身者，
次字首一碼取足者：



次字首
次字身

權
DTRG

複合字取碼：
字首取碼：



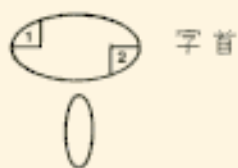
硫
MRYIU

次字首不能一碼取足者：

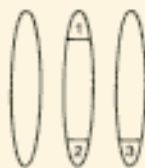


次字首
次字身

轉
JJJI



資
YOBUC



緞
VFHJE



建
NKLQ

重複字：x 為代表

日	A	目	XA	皮	DHE	板	XDHE
---	---	---	----	---	-----	---	------

難字取法：x 代表過為瑣屑部份

身	HKH	鷹	IXF	所	LXH	龜	NXU
井	TXC	慶	IXE	鹿	IXP	肱	LXL
龍	RXU	白	HX	鹿	LX	齊	YX

只取首尾的常用字形

門	AN	目	BU	鬼	HI	几	HU
鬥	LN	目	NL	佳	OG	气	OU
甦	VI	尸	YP	百	YR		

特殊字：木太火 的特殊組合

東	DW	束	DL	束	DWF	束	DB
末	DJ	束	HDL	本	DM	夷	KN
夷	KKKK	券	FQSH				

字身取碼：
字身為整體字者：



頭
MTMHC

Kangxi Radical Chart

Rad. No.	Kangxi Radical	Starts at Unicode	Rad. No.	Kangxi Radical	Starts at Unicode	Rad. No.	Kangxi Radical	Starts at Unicode
[1]	一	4E00	[2]	丨	4E28	[3]	丶	4E36
[4]	丿	4E3F	[5]	乙	4E59	[6]	乚	4E85
[7]	二	4E8C	[8]	亅	4EA0	[9]	人, 亻	4EBA
[10]	儿	513F	[11]	入	5165	[12]	八	516B
[13]	冂	5182	[14]	冃	5196	[15]	彡	51AB
[16]	几	51E0	[17]	凵	51F5	[18]	刀	5200
[19]	力	529B	[20]	勹	52F9	[21]	匕	5315
[22]	匚	531A	[23]	匸	5338	[24]	十	5341
[25]	卜	535C	[26]	凵	5369	[27]	厂	5382
[28]	厶	53B6	[29]	又	53C8	[30]	口	53E3
[31]	囗	56D7	[32]	土	571F	[33]	士	58EB
[34]	夂	5902	[35]	夂	590A	[36]	夕	5915
[37]	大	5927	[38]	女	5973	[39]	子, 孑, 孖	5B50
[40]	宀	5B80	[41]	寸	5BF8	[42]	小	5C0F
[43]	尢	5C22	[44]	尸	5C38	[45]	山	5C6E
[46]	山	5C72	[47]	川, 巛	5DDB	[48]	工	5DE5
[49]	己, 巳, 已	5DF1	[50]	巾	5DFE	[51]	干	5E72
[52]	彡, 彣	5E7A	[53]	广	5E7F	[54]	廴	5EF4
[55]	升	5EFE	[56]	弋	5F0B	[57]	弓	5F13
[58]	彡, 彣	5F50	[59]	彡	5F61	[60]	彡	5F73
[61]	心, 忄	5FC3	[62]	戈	6208	[63]	户, 户, 户	6236
[64]	手, 扌, 扌	624B	[65]	支	652F	[66]	支, 攴	6534
[67]	文	6587	[68]	斗	6597	[69]	斤	65A4
[70]	方	65B9	[71]	无	65E0	[72]	日	65E5
[73]	曰	66F0	[74]	月	6708	[75]	木	6728
[76]	欠	6B20	[77]	止	6B62	[78]	歹	6B79
[79]	夂	6BB3	[80]	母, 母	6BCB	[81]	比	6BD4
[82]	毛	6BDB	[83]	氏	6C0F	[84]	气	6C14

[85]	水, 氵	6C34	[86]	火, 灬	706B	[87]	爪	722A
[88]	父	7236	[89]	爻	723B	[90]	斗, 冂	723F
[91]	片	7247	[92]	牙	7529	[93]	牛	725B
[94]	犬, 犳	72AC	[95]	玄	7384	[96]	玉, 王	7389
[97]	瓜	74DC	[98]	瓦	74E6	[99]	甘	7518
[100]	生	751F	[101]	用, 甩	7528	[102]	田, 由, 甲	7530
[103]	疋	758B	[104]	疒	7592	[105]	𠂔	7676
[106]	白	767D	[107]	皮	76AE	[108]	皿	76BF
[109]	目	76EE	[110]	矛	77DB	[111]	矢	77E2
[112]	石	77F3	[113]	示, 礻	793A	[114]	肉	79B8
[115]	禾	79BE	[116]	穴	7A74	[117]	立	7ACB
[118]	竹	7AF9	[119]	米	7C73	[120]	糸, 纟	7CF8
[121]	缶	7F36	[122]	网	7F51	[123]	羊	7F8A
[124]	羽	7FBD	[125]	老, 耂	8001	[126]	而	800C
[127]	耒	8012	[128]	耳	8033	[129]	聿, 聿	807F
[130]	肉, 月	8089	[131]	臣	81E3	[132]	自	81EA
[133]	至	81F3	[134]	臼	81FC	[135]	舌	820C
[136]	舛	821B	[137]	舟	821F	[138]	艮	826E
[139]	色	8272	[140]	艸, 艹	8278	[141]	虍	864D
[142]	虫	866B	[143]	血	8840	[144]	行	884C
[145]	衣, 衤	8663	[146]	酉, 西	897E	[147]	見, 见	898B
[148]	角	89D2	[149]	言, 讠	8A00	[150]	谷	8C37
[151]	豆	8C46	[152]	豕	8C55	[153]	豸	8C78
[154]	貝, 贝	8C9D	[155]	赤	8D64	[156]	走	8D70
[157]	足	8DB3	[158]	身	8EAB	[159]	車, 车	8ECA
[160]	辛	8F9B	[161]	辰	8FB0	[162]	辶	8FB6
[163]	邑, 阝	9091	[164]	酉	9149	[165]	采	91C6
[166]	里	91CC	[167]	金, 钅	91D1	[168]	長, 长	9577
[169]	門, 门	9580	[170]	阜, 阡	961C	[171]	隶	96B6
[172]	隹	96B9	[173]	雨	96E8	[174]	青, 青	9751

[175]	非	975E	[176]	面	9762	[177]	革	9769
[178]	韋, 韦	97CB	[179]	韭	97ED	[180]	音	97F3
[181]	頁, 页	9801	[182]	風, 风	98A8	[183]	飛, 飞	98DB
[184]	食, 食	98DF	[185]	首	9996	[186]	香	9999
[187]	馬, 马	99AC	[188]	骨, 骨	9AA8	[189]	高	9AD8
[190]	髟	9ADF	[191]	門	9B25	[192]	鬯	9B2F
[193]	鬲	9B32	[194]	鬼	9B3C	[195]	魚, 鱼	9B5A
[196]	鳥, 鸟	9CE6	[197]	鹵	9E75	[198]	鹿	9E7F
[199]	麥, 麦	9EA5	[200]	麻	9EBB	[201]	黃, 黄	9EC3
[202]	黍	9ECD	[203]	黑, 黑	9ED1	[204]	黼	9EF9
[205]	黽, 龟	9EFD	[206]	鼎	9F0E	[207]	鼓	9F13
[208]	鼠, 鼠	9F20	[209]	鼻	9F3B	[210]	齊, 齐	9F4A
[211]	齒, 齿	9F52	[212]	龍, 龙	9F8D	[213]	龜, 龟	9F9F
[214]	龠	9FA0						

Japanese 外国語の勉強と教え

Keyboards

Once Japanese is selected as a language in WinCALIS Author or Student Workstation, five keyboard choices appear in the pull-down menu list. They are: Roman-to-Kanji, Roman-to-Kana Only, JIS Hiragana Keyboard, JIS Katakana Keyboard and Kanji by shape. Following are descriptions of each keyboard.

Roman-to-Kanji

This keyboard, which provides the user with the most efficient and complete access to Japanese characters, will transform Roman characters into kana or kanji. WinCALIS and WinCALIS Author support the two most common systems for transliterating Japanese in Roman characters: 1) the Hepburn system, and 2) the *Kunrei* system. Some Japanese word-processing systems allow many non-standard ways of entering Japanese in romanized form (e.g., “ca” for “ka,” “jya” for “ja/zya,” etc.). Refer to the Kana Chart to see how each character may be accessed in WinCALIS. Keep in mind also the peculiarities of transliterating Japanese, detailed in this section.

Roman-to-Kana Only

This keyboard lets you type in kana characters without going through the buffer, so you need not press Enter for each *bunsetsu* (refer to the Kana Chart to see how it accepts transliteration). It will automatically change Roman characters into hiragana; katakana may be accessed by pressing Shift or CapsLock.

JIS Kana Keyboards

These keyboards transform the QWERTY keyboard into one that directly enters kana characters, according to the Japanese Industrial Standard. For a display of the corresponding keyboard while typing in WinCALIS or WinCALIS Author, click on Help, and choose Keyboard Help.



Figure 35: Hiragana keyboard

To type a voiced kana (marked with *dakuten*, two dots), first type the key for its unmarked unvoiced counterpart, then press the overstrike key, F2, then press the *dakuten* accent key, which is the key marked with double quote “ (the shifted single quote ‘ key). See the chart for an example.

To type a kana beginning with the “p” sound (marked with *handakuten* circle), first type the key for its unmarked counterpart, then press the overstrike key, F2, then press the *handakuten* accent key, which is the “o” key (the letter “o” suggests the little circle diacritic). See the chart for an example.

Kanji by Shape

Kanji characters are composed of “root” shapes. Instead of retrieving a character by its pronunciation, this keyboard allows you to locate a character by its looks, by elements of its root structure. This is especially helpful in situations where you know what the character looks like, but not how it is

pronounced. The root structures are accessed by choosing the Kanji-by-shape keyboard, and clicking on the piano-keyboard-style buttons. Notice that the database requires you to preface some duplicate characters with the “X” key. To clear the Kanji by Shape Window, simply change your Language keyboard.

Peculiarities of Typing in Japanese

In Roman-to-Kanji keyboard:

<i>to get:</i>	<i>type:</i>
私	w-a-t-a-s-h-i
を	w-o
は	h-a
へ	h-e
いって	i-t-t-e
ほん	h-o-n-n
きょう	k-y-o-u
百	1-0-0 OR h-y-a-k-u
ABC	Ctrl + Shift + a b c
・ (中黒)	/
・ (bold period)	^

Particles

Particles must be entered by typing their literal romanization, that is, “w-o,” “h-a,” and “h-e.” The program won’t distinguish between the character when it is part of a word, or when it is a particle.

Typing Tiny Characters

WinCALIS will include the small characters within words with double consonants and modified syllables, for example, “i-t-t-e” or “s-h-u-u-m-a-t-s-u.” You may also get a small character by preceding the romanization with “x” or “l” (capitalize for katakana).

Typing Syllabic “n”

WinCALIS will include the “n” character within most words; the user may need to type an additional “n” within other words and always at the ends of words. So, WinCALIS will produce the proper hiragana if “k-o-n-b-a-n” is input, but notice you must type “h-o-n-n” to close the end of that word.

Long Vowels

In order to get the long “o” sound, the user must actually type in “u,” e.g.: “k-y-o-u.” For katakana, use the hyphen “-” for vowel extensions.

Typing in Katakana

To type in katakana, simply hold Shift or press CapsLock in the Roman-to-Kanji or Roman to Kana Only keyboard, or type in hiragana and then convert to katakana with the spacebar. As you may have noticed on the kana chart, there are a few unconventional combinations of characters. These will probably be used more for foreign words, e.g.: type in “f-o- -m-u” (include the hyphen for the long vowel). To convert, press the space bar, then Enter to accept the katakana conversion to the English word “form,” which is in the program dictionary (not in the internal dictionary). You must type in capitals to get proper nouns in katakana.

Numbers

WinCALIS will convert Arabic numbers to kanji. Simply type the digits while you are in the Roman-to-Kanji keyboard, press the spacebar to convert, then highlight and enter your choice of character(s).

The Active Buffer

There are important things to know about buffers when you are typing Japanese. If the characters are not black, you are in an active buffer and you must do something to clear the buffer. If the computer beeps at you, it means that you are trying to do something outside of the active buffer, and you must clear it (by pressing Enter) before you can continue.

When in the Roman-to-Kanji keyboard, this is the process for entering text:

- When you type hiragana, the letters will appear blue (active buffer). If you want the words to remain in hiragana, simply press Enter.
- If you want to convert the words into kanji, press the space bar, while the characters are still blue. Kanji will replace the words in blue and the sentence will appear red (active buffer). If all conversions are correct, press Enter (Note: The maximum length of a phrase to be converted is 32 characters. If you type the 32nd character, the phrase will automatically be converted).
- If any conversions are incorrect, do the following while the characters are still red. The reverse video block is your *bunsetsu*. Simply move the *bunsetsu* over the inappropriate conversion and press the Spacebar. The homophone list will appear on the screen. Move the arrows to the desired kanji, then press Enter. The selected kanji will appear in the phrase in reverse video. If this is correct, press Enter again.
- WinCALIS divides your sentence into *bunsetsu* blocks as its dictionary sees fit, but you may change these blocks. First, convert back to hiragana by choosing it from the list (if you try to change the *bunsetsu* while there is still Kanji within the active buffer, you will get a pop up message reminding you to convert). Use the shifted left arrow to mark the first *bunsetsu*. At this point, you may convert that block by pressing the spacebar, or use the arrow keys to “jump” from block to block. To continue to change *bunsetsu* blocking, move to the block that you would like to adjust, and use the shifted right and left arrow keys to shrink or expand it.
- The user may find it helpful to clear the buffer (press Enter) early and often while working in the Roman-to-Kanji keyboard so the *bunsetsu* stay manageable. This will be convenient both for maintaining kana, and in locating the desired kanji. If the homophone list does not display the character you want, try changing the blocking of the *bunsetsu*. With use, you will become more familiar with how the transliteration process occurs within WinCALIS.

Look at the following passage, and notice the *bunsetsu* blocks—the bars indicate the points where conversions may be most efficiently placed (where the spacebar should be pressed):

The Way of the Bunsetsu

「ユーザーフレンドリ」 | と | 呼ばれる |
理由は、ごく | 普通の | 人がごく |
簡単に | 使用 | 方法を | マスター | でき、 |
毎日の | タイプ | の | 仕事にすぐ |
役立てることができることからきている。 |

This passage includes hiragana, katakana, kanji, and punctuation, so it is a nice exercise routine for typing Japanese. Notice how each bunsetsu is entered:

よばれる	—	呼ばれる
りゅうは、ごく	—	理由は、ごく
ふつうの	—	普通の
ひとがごく	—	人がごく
かんたんに	—	簡単に
しょう	—	使用
ほうほうを	—	方法を
まいにちの	—	毎日の
しごとにすぐ	—	仕事にすぐ
やくだてることができることからきている。—		
役立てることができることからきている。		

The Homophone List

When you choose a kanji from the homophone list, the program will “remember” that one and present it next time the same kana are entered. The kanji most frequently used will be given priority over the other homophones (this does not rearrange the homophone list, which is constant. Rather, a separate user history file is being created and constantly updated).

Typing in Two Languages

Toggling between two languages can be done either by: clicking and dragging the mouse on the language choice from the options list; pressing F3 to toggle to the language at the top of the language list (usually English); or by clicking and dragging with the right-hand mouse button.

Punctuation in Japanese

Setting aside keyboard exploration time may make entering Japanese text easier. Because Japanese uses some punctuation that is exclusive to Japanese, the characters may be hidden from view on the

standard English keyboard. Most of the punctuation parallels the basic keyboard, but it is in a *zenkaku*, full-width form. The keyboard layout is displayed by clicking on Help, Keyboard Help. See the chart for a sample of available marks.

Kana Kode Key

In order to obtain a kana character below, you must type in one set of Roman letter(s) listed beneath it. The first line of rōmaji is the *Kunrei* system. Where applicable, the second line is the Hepburn system, and the third is nonstandard. Keep in mind the peculiarities of typing in Japanese, detailed in this section.

あ	い	う	え	お				
a	i	u	e	o				
あ	い	う	え	お				
xa	xi	xu	xe	xo				
la	li	lu	le	lo				
か	き	く	け	こ	きや	きゆ	きよ	
ka	ki	ku	ke	ko	kya	kyu	kyo	
			co					
さ	し	す	せ	そ	しや	しゆ	しえ	しよ
sa	si	su	se	so	sya	syu		syo
	shi	ce			sha	shu	she	sho
	ci							
た	ち	つ	て	と	ちや	ちゆ	ちえ	ちよ
ta	ti	tu	te	to	tya	tyu		tyo
	chi	tsu			cha	chu	che	cho
な	に	ぬ	ね	の	にや	にゆ		によ
na	ni	nu	ne	no	nya	nyu		nyo

は	ひ	ふ	へ	ほ	ひゃ	ひゅ	ひょ
ha	hi	hu fu	he	ho	hya	hyu	hyo
					ふあ	ふい	ふえ
					fa	fi	fe
ま	み	む	め	も	みゃ	みゅ	みょ
ma	mi	mu	me	mo	mya	myu	myo
や	(い)	ゆ	(え)	よ	ゃ	ゅ	ょ
ya		yu		yo	xya lya	xyu lyu	xyo lyo
ら	り	る	れ	ろ	りゃ	りゅ	りょ
ra	ri	ru	re	ro	rya	ryu	ryo
わ	ゐ	(う)	ゑ	を	わ	ヰ	ヱ
wa	wi		we	wo	Wa	WI	WE
ん		う			っ	カ	ケ
nn		vu			xtu	XKA	XKE
n'					ltu	LKA	LKE
					xtsu		
が	ぎ	ぐ	げ	ご	ぎゃ	ぎゅ	ぎょ
ga	gi	gu	ge	go	gya	gyu	gyo
					ぐわ		ぐえ
					gwa		gwe
ざ	じ	ず	ぜ	ぞ	じゃ	じゅ	じょ
za	zi	zu	ze	zo	zya	zyu	zye
	ji				ja	ju	je
だ	ぢ	づ	で	ど	ぢゃ	ぢい	ぢゅ
da	di	du	de	do	dya	dye	dye
		dzu					
ば	び	ぶ	べ	ぼ	びゃ	びゅ	びょ
ba	bi	bu	be	bo	bya	byu	byo
ぱ	ぴ	ぷ	ぺ	ぽ	ぴゃ	ぴゅ	ぴょ
pa	pi	pu	pe	po	pya	pyu	pyo
ワ	ウィ	ウ	ウェ	ウオ	ヴァ	ヴィ	ヴ
wa	uxi	u	uxe	uxo	VA	VI	VU
	uli		ule	ulo			VE
							VO

Full-width Characters

Full width Roman characters are accessed by typing Ctrl + Shift + the character. This provides aesthetic appeal in situations where Roman characters are included within Japanese text.

Spelling Variation In Japanese

Note the potential for confusion on the student's part (in the WinCALIS language instruction system) because of spelling variation allowable in written Japanese. The careful courseware author will anticipate alternative spellings along the lines of those listed below in writing correct and incorrect responses.

Preferred form	Alternate acceptable form 1	Alternate form 2 (questionable)
あさ	朝	
あした	明日	あす、みょうにち
後で	あとで	
現れる	表れる	
ある	有る	在る
いい	良い	善い、佳い、好い
行く	いく	ゆく
いたす	致す	
いる	居る	
いる	要る	
うち	内	
映る	写る	
おもしろい	面白い	
代える	替える	換える
かえる	帰る	返る

かた	方	
かれ	彼	
河	川	
きのう	昨日	さくじつ
きょう	今日	
具合	ぐあい	具合い
ください	下さい	
ぐらい	位	くらい
来る	くる	
けさ	今朝	
ご、お	今御	
ごく	極	極く
こんや	今夜	
こと	今事	
ころ、ごろ	頃	
才	歳	
しだい	次第	
ずいぶん	随分	
誰	だれ	
つける	付ける	
できる	出来る	
という	と言う	
等	など	とう
時	とき	
所	ところ	
ない	無い	
なに	何	
なにか	何か	
なる	成る	
なる	爲る	

そのば	その場		
ばあい	場合		
はいる	入る		
初めて	始めて		
放す	離す		
早い	速い		
ひとつ	一つ		
ひとり	一人		
ほう	方		
他	ほか	た	
本当に	ほんとうに		
まいる	参る		
もうしわけ	申し訳		
もとづく	基づく		
もの	物		
よろこぶ	悦ぶ	喜ぶ	慶ぶ
わかる	解る	分かる	
わけ	訳		
わたし、わたくし	私		

Note: The path to the “history” file, which stores your most recent choice among homophone alternatives, “wcjpnhst.dic,” can be redefined by adding the following line to the [WinCALIS] section of the main initialization file “WinCALIS.ini”:

```
Jpnhistory=c:\wincalis\
```

Additional Notes for Authors:

Authoring a Japanese lesson can be done, for the most part, according to the manual. There are, however, a few variations and considerations that arise because of the integration of two different languages and four different alphabets. In addition to keeping in mind all of the quirks of typing in Japanese, the author must consider how the text content is processed and presented to the student user.

Preparing Students for the Mechanics of the Lesson

In the Popup Directions Window the author may want to tell the student more than just “Fill in the blanks.” The students may need reminding on how to access different keyboards, what keyboard they should have for the lesson, and any other information that would help them in the mechanics of the lesson. Remember that in the Roman-to-Kanji keyboard, the student must

- Be in the right keyboard
- Type in an answer
- If necessary, choose a kanji from the homophone list

-
- Clear the active buffer (by pressing Enter)
 - Then, finally: Enter the answer

“Cloze Exercise” Hints

WinCALIS is very specific about what is marked within the text of a cloze exercise, so it is also very specific about what the student enters. Since Japanese may be entered by different methods (for example, the roman “w-a-t-a-k-u-s-h-i” converts into one single Kanji character) and the different ways call for different answer analyses, the author should select the Enable Fixed Blank Length option so that the student is able to type in the answer. The fixed blank length that you choose must be long enough to accommodate the longest string that will be typed in (example: 9 for "watakushi"). The author should also let the student know what sort of response may be expected for any particular task (Roman characters only? Kana only? Kanji?)

Wild Card Hints

Wild card are comprised of a specific set of characters. Notice that these characters must be reproduced exactly if they are to be recognized as wild cards. For example, the ampersand ‘&’, the asterisk ‘*’, the bar ‘|’, and the curly braces ‘{ }’ must be entered in this ANSI form – not the full-width form sized for Japanese typesetting.

Korean

How to Type Hangul

Select one of the Hangul keyboards in the "Korean" submenu of the "Language" menu. There are four different Hangul keyboards to choose from: the Phonetic keyboard, the Standard keyboard, the Kong keyboard, and the Morse keyboard.

The Hangul (Phonetic) Keyboard

The Phonetic keyboard is the typical student keyboard. The consonants and the vowels are arranged to reflect the similar sounds of the Roman alphabet. For example, the consonant **ㅍ** is located on the "p" key, and the vowel **ㅏ** is located on the "a" key. Thus, it may be the easiest keyboard for a native English speaker to use. However, some Korean letters do not have an equivalent in English, and thus are placed on unused keys on the keyboard. For example, **ㅇ** is located on the "q" key with no relevance to sound (but a mnemonic association with the shape of the letter "Q"). Likewise, **ㅎ** looks a little like a "Z" on its side, so it is on the "z" key; and **ㅡ**, which is romanized **ŭ** (a "double u") is on the "w" key. The user must become familiar with these "exception" letters.

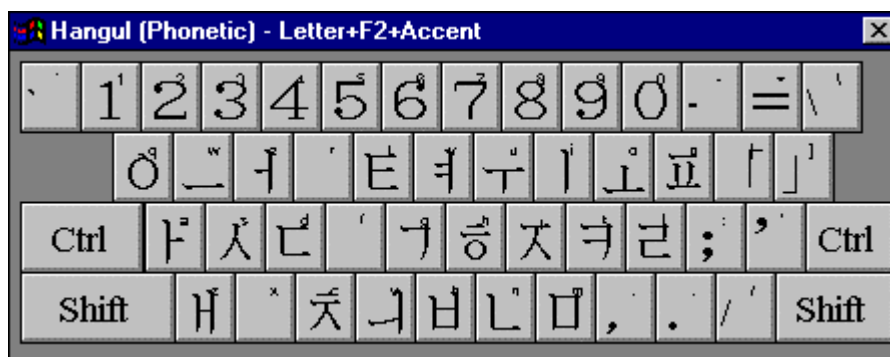


Figure 36: Korean Phonetic keyboard (unshifted)

The Hangul (Standard) Keyboard

The Standard keyboard, as suggested by its name, is the standard keyboard in Korea (the Korean typewriter keyboard). Thus, it is probably the keyboard that a native Korean would be familiar with. It is divided into two parts: The vowels are located on the right half of the keyboard, and the consonants are located on the left half of the keyboard.



Figure 37: Korean Standard keyboard (unshifted)

The Hangul (Kong) Keyboard

The Kong keyboard is separated into three parts: The initial letters (consonants, such as ㄱ, ㅋ, ㆁ, etc.) are placed on the right third of the keyboard. The medial letters (vowels such as ㅏ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㅣ, etc.) are placed on the middle third of the keyboard. The final letters (consonants) are on the left third of the keyboard. The user must be aware of these differences. For example, although the initial and the final consonants look the same (ㄱ, an initial on the "h" key, and ㄱ, a final on the "s" key), they cannot be used interchangeably.



Figure 38: Korean Kong keyboard (unshifted)

The Hangul (Morse) Keyboard

The Morse keyboard, among the four, requires the greatest number of keystrokes to type. There are no compound vowels, only simple. Thus, the user must combine the simple vowels to make compound vowels. For example, ㅏ + ㅑ = ㅓ. In addition, the vowels and consonants are scattered about the keyboard in relation to frequency of use and convenience of layout of the particular letter (similar to the arrangement of the English keyboard).



Figure 39: Korean Morse keyboard (unshifted)

Hangul-Hanja Conversion

Hangul-to-Hanja (F8)

To convert the syllable or word to the left of the insertion point (the blinking caret) written in Hangul into its Hanja equivalent, press the F8 function key, or click on the menu item Tools, Language Options, Korean Options, Hangul-to-Hanja. (One of the Korean language keyboards must be the currently selected keyboard.) The conversion engine scans to the left from the insertion point attempting to match up to four Hangul characters. If there are homophone equivalents, they will appear in a Selection Window. Choose the desired equivalent.

For example, to convert 도서관 to 圖書館, press F8 with the insertion point to the right of 도서관.

Hanja-to-Hangul (F7)

To convert the syllable or word to the left of the insertion point (the blinking caret) written in Hanja into its Hangul equivalent, press the F7 function key, or click on the menu item Tools, Language Options, Korean Options, Hanja-to-Hangul. (One of the Korean language keyboards must be the currently selected keyboard.) The conversion engine scans to the left from the insertion point attempting to match up to four Hanja characters. If there is more than one Hangul equivalent (i.e., there is more than one reading for the Hanja), they will appear in a Selection Window. Choose the desired equivalent.

For example, to convert 韓國 to 한국, press F7 with the insertion point to the right of 韓國.

You can also enter Korean Hanja characters through two other methods:

Shape-based coding: The Hanja By Shape Keyboard

Under the Language or L:[Language] menu, select the "Hanja by shape" keyboard in the Korean submenu. For details on this method see "The Cangjie Input Method". Cangjie, the legendary inventor of writing in ancient China, is known in Korean as Ch'anggyôl. To clear the Cangjie Window, simply change your Language keyboard.



The Hangu (Morse) Keyboard

Korean Transliteration - Letter+F2+Accent

'	1 ¹	2 ²	3 ³	4 ⁴	5 ⁵	6 ⁶	7 ⁷	8 ⁸	9 ⁹	0 ⁰	-	=	\
ö ^q	w ^w	e ^e	r ^r	t ^t	y ^y	ü ^u	i ⁱ	ö ^o	p ^p	[]		
Ctrl	a ^a	s ^s	d ^d	ö ^r	g ^g	h ^h	j ^j	k ^k	l ^l	;	'	Ctrl	
Shift	z ^z	ü ^x	c ^c	ü ^v	b ^b	n ⁿ	m ^m	,	.	/	Shift		

Figure 41: Korean Transliteration keyboard

Chapter 10: Configuring and Customizing WinCALIS and WinCALIS Author

These are various topics related to the general subject of customizing your WinCALIS Author Workstation so it serves you the best, maximizing your authoring productivity.

<i>To do this...</i>	<i>See the section:</i>
Learn how to set the font size for use in your WinCALIS Author, set default directories, change the virtual keyboard picture, change the Language Menu, and much more	The Main WinCALIS Initialization File WINCALIS.INI
Learn how to change or fine-tune the default font(s) for use in your WinCALIS Author	The Default Character-to-Font Mapping File WCFNT.INF
Learn about how to make your own language keyboard or modify an existing one	Defining Language Keyboard Files
Learn about how the Unicode-smart WinCALIS 2.x system globally defines punctuation, capitalization, and accented characters	The WinCALIS Global Information File WINCALIS.WGI
Learn how to create, modify, and install on-line dictionaries to accompany your lessons	User-Supplied On-line Dictionaries
Learn more about WinCALIS fonts and typefaces	Fonts
To find out about setting up a Spelling Checker for you or your students	Linking to a Spelling Checker
To find out which WinCALIS files are essential and which you can get rid of without worry	Deleting Nonessential Files to Save on Disk Space
Learn about making the temperamental Windows Media Control Interface (MCI) work right	Installing MCI Drivers For Multimedia
For tips on network installation and operation of WinCALIS	Installing WINCALIS on a Network
To find out how to remove the WinCALIS Author Workstation from your system	How to Uninstall WinCALIS and WinCALIS Author

The Main WinCALIS Initialization File WINCALIS.INI

General Tips

The main WinCALIS initialization file WINCALIS.INI, which contains all kinds of customizable settings, is to be found in the main WinCALIS program directory (usually the parent directory of the default scripts directory).--e.g., C:\WINCALIS. It is created dynamically at the time of installation. It

is a typical ASCII Windows *.INI file. It can be edited with a plain-text editor like the Windows Notepad. Be sure to use the Windows File Manager or other tool to make a backup copy of WINCALIS.INI, before you make any changes in it.

If you are currently running WinCALIS or WinCALIS Author, you will have to exit and restart Author and WinCALIS for any new settings to take effect.

If you wish to make only a temporary change to one or another setting in your WINCALIS.INI file, you can leave the Notepad running in the background, with the WINCALIS.INI file still open in it. Then after you have finished using Author, you can return to the Notepad and change the setting back again.

Changing the Default Font Size for Use in WinCALIS Author

(For how to change the font or typeface itself, see The Default Character-to-Font Mapping File WCFNT.INF.)

WinCALIS Author always displays text in “Medium” font size. But it makes it much easier to create “Small” font size Text or Tutorial Windows, or any non-Medium windows, when Author itself is set to the same pixel height. You can do this yourself in a few minutes! If you are going to spend more than a half hour typing, editing, and formatting text to go into Text or Tutorial windows, it is probably worth your effort.

Here's how to do it. WinCALIS Author will still use “Medium” for any font it needs. But you can redefine the value of “Medium” to any pixel height you want.

The actual point size of the “Medium” used by WinCALIS Author is determined by the pixel height settings in the default language .INI file defined in the “DefaultIni=” line in the [WinCALIS] section of WINCALIS.INI. The default language .INI file is set for the United States and Western Europe to “LATIN1.INI.” Changing the line “Medium=21” in the [FontSize] section of “LATIN1.INI” to some greater or smaller pixel height than 21 pixels will adjust the size of the default font accordingly in WinCALIS Author and the Author Workstation's copy of WinCALIS. All pixel heights are set by default for standard VGA resolution (640x480 pixels) as follows:

```
Tiny=12
Fine=16
Small=19
Medium=21
Large=32
Xlarge=48
Huge=66.
```

So to change the font size for ANSI (all West European languages), edit the LATIN1.INI file. (The procedure is identical for other character sets--Cyrillic, Greek, Hebrew, Thai, etc.)

In order to make the Presentation (Text) Window (and every other window) in WinCALIS Author (always “Medium”) look exactly like the WinCALIS Presentation (Text) Window or a Tutorial Window (“Small” by default), temporarily set the value of “Medium” to the same as the value of “Small”:

```
Medium=19
```

Or, in order to make the Interactive (Question) Window (and every other window) in WinCALIS Author (always “Medium”) look exactly like a WinCALIS Interactive (Question) Window which has been set to Extra Large font size with a <SET FONT XLARGE> command, temporarily set the value of “Medium” to the same as the value of “Xlarge”:

Medium=48.

Then exit both WinCALIS and WinCALIS Author and restart WinCALIS Author.

You can leave the LATIN1.INI file running in Notepad in the background. Then when you no longer need a smaller or larger font size, restore the original settings and save the file again.

Special versions of “latin1.ini” for use with EGA monitors and with VGA monitors at higher than standard VGA resolution are distributed as part of the WinCALIS Author and Student Workstation packages.

The pixel heights for the fonts in many other character sets can be adjusted in the same way, by specifying the pixel heights in the language keyboard *.INI file you will first use to type that character set, generating that font on your screen.

Setting the Virtual Keyboard Type

The virtual keyboard picture displayed when Keyboard Help is selected is defined in the WINCALIS.INI file in the main WinCALIS subdirectory. It can be redefined by carefully editing the line in the [WinCALIS] section

KeyboardType=0

Keyboard arrangements are perhaps best described by the position of the backslash (reverse solidus) key (at least for U.S. keyboards). Substitute a number corresponding to one of the following available pictures:

- Type 0: Backslash key (\) on top row, next to Backspace key (default)
- Type 1: Backslash key (\) on 2nd row from top, below Backspace key
- Type 2: Backslash key (\) on bottom row at extreme right, below Enter key
- Type 3: Backslash key (\) on bottom row at extreme left

Setting the WinCALIS Startup Mode (*.CAL or *.WCL)

WinCALIS is set to start up on your Author Workstation showing the text version, rather than the binary distribution version, of lesson scripts, often referred to by their file extension as *.CAL files. You can change this default setting so that WinCALIS starts up on your Author Workstation in “No Conversion (*.WCL)” mode (the only mode on Student Workstations).

Use the Windows Notepad, the DOS Editor, or other plain-text editor to edit the “ListBoxDisplay=” line in the [WinCALIS] section of WINCALIS.INI. The values are as follows:

ListBoxDisplay=0	(sets No Conversion *.WCL mode)
ListBoxDisplay=1	(sets Text Conversion *.CAL mode).

Note: The option to switch between “No conversion” and “Text conversion” is found only on Author Workstation versions of WinCALIS.

Setting WinCALIS Window Defaults

You may specify the aesthetic qualities of the screen in each of the various windows that WinCALIS employs on *your workstation only*. For each window that WinCALIS utilizes, you may specify the text

color, text attributes (e.g. **bold**, *italic*, etc.), point size, and background color. Each WinCALIS window (Directions, Text, etc.) may have different characteristics. These are determined by settings in the [WinCALIS] section of your WINCALIS.INI file. You can change these by clicking on the Options menu in the WinCALIS opening screen and selecting the menu item Window Defaults. Then make your selection for each window in the pull-down combo boxes. When you press the “OK” button, your preferences will be saved in your WINCALIS.INI file. (The “Language” box is not relevant to WinCALIS 2.x and is currently ignored.)

This is how the lines appear in WINCALIS.INI with the out-of-the-box default values (Normal, Black on White Text):

```
[WinCALIS]
.
.
Question=Times Medium 0 BLACK WHITE
Text=Times Small 0 BLACK WHITE
Lesson_Description=Times Medium 0 BLACK WHITE
Directions=Times Medium 0 BLACK WHITE
Tutorial=Times Small 0 BLACK WHITE
Diagnostic=Times Small 0 BLACK WHITE
```

Note: The option to set “Window defaults” is found only on Author Workstation versions of WinCALIS.

Changing the WinCALIS Author Desktop

You may have noticed that WinCALIS Author saves your “desktop”—that is, the number and arrangement of open windows, along with certain other settings—which you have established while editing a lesson as part of the *.WCA database file for that lesson. When you open an existing *.WCA file, the desktop will return to how you left it when you last saved the lesson.

In the Desktop submenu under the Window menu you can “Save” the current desktop arrangement, “Restore” the previous arrangement, or go back to the “Default” arrangement, as it was set in WINCALIS.INI during installation of Author Workstation .. Here's how each of these sections looked in your WINCALIS.INI out of the box:

```
[Initial]
Word=Yes
Tone=Yes
Def=No
NoDisk=Yes
Simple=Yes
NoConvrt=No
NoCache=No
Net=Yes
InitialLanguage=English/W.Europe
Question=SHOW 316 181 316 181
Directions=SHOW 0 181 316 181
Text=SHOW 316 0 316 181
Lesson_Description=SHOW 0 0 316 181
Scratchpad=ICON
Pass_Through=ICON
Active=Question
Wordwrap=No
Statusbar=Yes
```

```
[Default]
```

```
Word=Yes
Tone=Yes
Def=No
NoDisk=Yes
Simple=Yes
NoConv=No
NoCache=No
Net=Yes
InitialLanguage=English/W.Europe
Question=SHOW 316 181 316 181
Directions=SHOW 0 181 316 181
Text=SHOW 316 0 316 181
Lesson_Description=SHOW 0 0 316 181
Scratchpad=ICON
Pass_Through=ICON
Active=Question
Wordwrap=No
Statusbar=Yes
```

The first eight lines in each section refer to Chinese internal dictionary search options, which are also a part of the desktop configuration.

Setting the Top and Initial Languages in your Language Menus

The “top” language in your WinCALIS Author L: menu and WinCALIS Language menu is determined by the default language .INI file defined in the “DefaultIni=” line in the [WinCALIS] section of WINCALIS.INI. The default language .INI file is set by default for the United States and Western Europe to “LATIN1.INI”. If the default .INI file is changed to some other language .INI file, the “top” language will also be changed accordingly.

The initial language for use on startup in WinCALIS and WinCALIS Author (also by default set to “English/W.Europe”) is also customizable by the courseware author or WinCALIS system administrator. The initial language is determined by the setting of the InitialLanguage= line in the [Initial] section of the WINCALIS.INI file. The out-of-the-box value for this line is InitialLanguage=English/W.Europe.

Rearranging the Language Menus

The list of languages and keyboards in the L:[Language] menu in WinCALIS Author and the Language menu in WinCALIS is constructed dynamically out of the language keyboard *.INI and *.WCK files present in the WinCALIS program directory each time you run either of these programs.

The list is arranged into language or script grouping submenus, each one containing various language keyboards, based on the list of Submenus and *.INI files in the [WinCALIS] section of WINCALIS.INI reproduced below.

A typical Language Menu-related section of WINCALIS.INI

```
[WinCALIS]
.
.
.
SubMenu1=Arabic script
SubMenu2=African
SubMenu3=Asian (other)
SubMenu4=Chinese
SubMenu5=Europe (other)
SubMenu6=Japanese
SubMenu7=Special Fonts
SubMenu8=Cyrillic script
```

SubMenu9=Hebrew script	Language570=KOREANM
SubMenu10=Korean	Language580=KOREANS
SubMenu11=Indic scripts	Language590=KOREANU
Language20=AMHARIC	Language600=KORHANJA
Language60=ARMENIAN	Language640=LATIN
Language30=ARABICC	Language650=LATVIAN
Language50=ARABICX	Language660=LITHUAN
Language40=ARABUSG	Language680=MACEDON
Language90=BELARUS	Language450=MAGYAR
Language140=BOPOMOFO	Language700=MONGOL
Language100=BULGARIA	Language710=PASHTO
Language150=CANGJIE	Language720=PERSIANC
Language120=CENTASIA	Language730=PERSIANP
Language130=CHINESE	Language740=PERSIANX
Language180=CROATIAN	Language750=POLISH
Language190=CZECH	Language760=PORTUG
Language220=DUTCH	Language490=ROMAJI
Language230=EEUR	Language500=ROMAJINC
Language240=ENGLISH	Language810=ROMANIAN
Language250=FORMSYMB	Language780=RUSSIANC
Language260=FRENCH	Language790=RUSSIANS
Language280=GEORGIAN	Language800=RUSSTECH
Language300=GERMAN	Language820=SCANDIN
Language320=GREEKCAL	Language830=SERBIAN
Language330=GREEKPC	Language840=SINDHI
Language360=HAUSA	Language850=SLOVAK
Language400=HEBREWI	Language870=SPANISH
Language420=HEBREWK	Language890=TAGALOG
Language430=HEBREWT	Language920=THAI
Language410=HEBREWU	Language930=TRANSLIT
Language510=HIRAGANA	Language940=TURKISH
Language460=IBM-PC	Language950=UKRAINE
Language470=IPA	Language960=URDU
Language480=ITALIAN	Language970=VIETNAME
Language540=KANJICJ	Language980=YIDDISH
Language530=KATAKANA	Language990=YORUBA
Language560=KOREANK	

For example, the line “Language100=BULGARIA” tells WinCALIS or WinCALIS Author to look for a file named BULGARIA.INI. Its sequence in the list of ascending numbers “Language#” more or less determines the alphabetical sequence in the language menu or submenu of the language keyboard identified within BULGARIA.INI, which happens to be Bulgarian.

WinCALIS or WinCALIS Author then looks in the file BULGARIA.INI. It looks like this:

*A typical language keyboard *.INI file*

```
[Language]
Name=Bulgarian
Face=Times
SubMenuID=8
RightLeft=0
CharSet=0
Keyboard=Bulgaria
LanguageID=100
Pitch=2
```

```
DeadKey=  
Font1=Kiev  
Font2=Moscow  
Font3=Sofia
```

```
[FontSize]  
Tiny=12  
Fine=16  
Small=19  
Medium=21  
Large=32  
ExtraLarge=48  
Huge=66
```

The “Name=Bulgarian” line determines the menu listing, which will be just that, “Bulgarian,” and the SubMenuID=8 line places the listing for Bulgarian in the “Cyrillic script” subdirectory.

WinCALIS or WinCALIS Author then gets the name of the *.WCK keyboard file from the “Keyboard=Bulgaria” line and looks for the file BULGARIA.WCK. If it finds such a file, it will fully enable the menu listing. If it fails to find the file, it will display the menu listing for Bulgarian dimmed.

The lines Font1=Kiev, Font2=Moscow, and Font3=Sofia are informational only. These are the font names which are displayed in the Font dialog box accessed from the Format menu in WinCALIS Author, and which Author will insert in <SET FONT facename> commands for you semi-automatically.

For the ANSI character set (Unicode page 0) you can freely add as many fonts here as are available (and which you believe will also be available to students working your lessons). For other character sets, fonts must be consistent with the WinCALIS Unicode-character-to-font mapping tables UNI2FNT1.WCF....UNI2FNT6.WCF. Fonts designed for use in non-Unicode compliant systems, such as Windows 3.1 word processing programs, will not work with WinCALIS. If you have questions, feel free to contact the Humanities Computing Laboratory or its agents.

You can rearrange your own Language menus as you please. For example, you can promote your favorite language keyboard to membership in the main menu by removing the SubMenuID= line, or just “commenting it out” with a semicolon thus:

```
;SubMenuID=8
```

and you can move it up or down the menu by changing its LanguageID#.

The [FontSize] section of the language keyboard *.INI file determines the actual height in pixels of each font size used by WinCALIS or WinCALIS Author. See Changing the Default Font Size for Use in WinCALIS Author.

(The lines Face=, RightLeft=, CharSet=, Pitch=, and DeadKey= are no longer used in the WinCALIS 2.x system.)

Setting Default Directories

Setting the Conversions Subdirectory

The “CONVERT” subdirectory holds code conversion utilities and filters, if during Setup you selected the option to install “Code Conversion Utilities.” If you did not select this option and you wish to

convert files from one coding standard to another, you will need to run Setup again and select this option to install the necessary files.

The path to the “CONVERT” subdirectory is defined in the WINCALIS.INI file in the main WinCALIS subdirectory. It can be redefined to whatever path you choose by carefully editing the line in the [WinCALIS] section

Conversions= ...

Setting the Documents Subdirectory

The “DOCS” subdirectory can be used for your document text files, as opposed to WinCALIS scripts. It is not essential to the operation of WinCALIS programs and may be deleted if you wish.

The path to the “DOCS” subdirectory can be redefined to whatever path you choose by carefully editing the line in the [CalAuth] section

Documents= ...

Setting the Graphics Subdirectory

The “GRAPHICS” subdirectory can be used to hold *.BMP bitmap graphics files. It is particularly useful when such files are shared by multiple scripts in different directories.

The path to the “GRAPHICS” subdirectory can be redefined to whatever path you choose by carefully editing the line in the [WinCALIS] section

Graphics= ...

When a script requires a *.BMP file, WinCALIS first checks the directory defined by the Graphics= line in WINCALIS.INI, then checks the “current” directory (the directory in which the *.WCL script file resides) or the path specified in the <GRAPHIC> command in the script.

Setting the “Home” Directory

You will not ordinarily need to change the program directory setting. But if you need to move the entire WinCALIS program directory and all its subdirectories to another drive or directory, you will then need to change the “Home=” line in WINCALIS.INI, along with a number of other lines. If WinCALIS was installed in the default directory location “WINCALIS” on drive C:, the “Home=” line would look like this.

Home=C:\WINCALIS\

It is also important for network installations of WinCALIS to be aware of the drive letter for the “Home” and other directories in WINCALIS.INI. For example, it is sometimes the case with installation on a network server that the DOS drive letter by which that server is known to a local workstation changes. If that happens, WinCALIS will not function properly on the local workstation. See Installing WinCALIS on a Network.

Be careful in editing the “Home=” line to retain the final backslash (reverse solidus).

Setting the Multimedia Subdirectory

The “MLTMEDIA” subdirectory can be used to hold multimedia resources, such as *.WAV audio files, *.AVI digital video clips, MIDI audio files, or animation files. It is particularly useful when such files are shared by multiple scripts in different directories.

The path to the “MLTMEDIA” subdirectory can be redefined to whatever path you choose by carefully editing the line in the [WinCALIS] section

Multimedia= ...

When a script requires a multimedia file, WinCALIS first checks the directory defined by the Multimedia= line in WINCALIS.INI, then checks the “current” directory (the directory in which the *.WCL script file resides), then checks for any DOS path to the file defined in the *.MMD file for that script.

Setting the Records Subdirectory

The “RECORDS” subdirectory can be used to hold the Borland Paradox database tables which are generated by WinCALIS 2.x to hold student records, when record-keeping is installed.

The path to the “RECORDS” subdirectory can be redefined to whatever path you choose by carefully editing the line in the [WinCALIS] section

Records= ...

Setting the Scripts Subdirectory

The “SCRIPTS” subdirectory is the directory in which both WinCALIS and WinCALIS Author by default first look for lessons when they start up, as installed during Setup.

WinCALIS and WinCALIS Author can each have their own default scripts directory. The path to the “SCRIPTS” subdirectory is defined for WinCALIS in the [WinCALIS] section of WINCALIS.INI and for WinCALIS Author, in the [CalAuth] section of WINCALIS.INI. Each of these can be redefined to whatever path you choose by carefully editing either of the lines

Scripts= ...

Setting the Dictionaries Subdirectory

The “DICTS” subdirectory holds any preinstalled on-line dictionaries.

The path to the “DICTS” subdirectory is defined in the WINCALIS.INI file in the main WinCALIS subdirectory. It can be redefined to whatever path you choose by carefully editing the line in the [WinCALIS] section

Dictionaries= ...

See Also: How to Create Your Own On-line Dictionary
Language Keyboard .INI Files

Linking to a Spelling Checker

The WinCALIS Author Workstation Setup package includes a shareware Windows spelling checker program, WinSpell 3.0®, which may be freely distributed.

WinSpell or another user-supplied spelling checker can be “linked” to WinCALIS and WinCALIS Author menus by specifying the program name and the DOS path to it in the “SpellChecker=” line in the [WinCALIS] section of WINCALIS.INI. For example:

SpellChecker=c:\wspell\wspell.exe

will enable the “Check Spelling” menu item in the WinCALIS and WinCALIS Author “Tools” menu. If that menu item is selected, the WinSpell program will be executed, as if it had been executed by the <EXEC: WINSPELL.EXE> command in WinCALIS.

The Default Character-to-Font Mapping File WCFNT.INF

Both WinCALIS Author and WinCALIS always display text in the typeface defined in the file “WCFNT.INF” in the WinCALIS program directory. This typeface is by default “Times New Roman” for the ANSI character set (Unicode page 00) and extended Latin character set (Unicode pages 01-02, etc.). The default typeface for use in Author and the Author Workstation's copy of WinCALIS may be changed to Arial, Courier New, Fixedsys, MS Sans Serif (the font you are reading in now! -- a raster screen font which is preferred to TrueType fonts for Windows Help because of its superior legibility at low resolutions such as on computer monitors), or any other font available to the courseware author.

Just edit the file “WCFNT.INF” with a plain text editor, such as the Windows Notepad. (Keep a backup copy of the original file.) For example, to change the font for just the basic Windows ANSI (Latin1) character set to Arial, the first line in the [FaceNames] section should be changed to read:

00=Arial.

In a similar fashion the default font for any character set may be changed by editing the line(s) for its corresponding code pages in “WCFNT.INF.”

If you are currently running WinCALIS or WinCALIS Author, you will have to exit and restart Author and WinCALIS for the new settings to take effect.

If you wish to make only a temporary change in your Author font, you can leave the Notepad with running in the background, with the WCFNT.INF file still open in it. Then after you have finished using Author, you can return to the Notepad and change the font back again.

Warning: While the settings in your local “WCFNT.INF” determine the default font for your Author Workstation, they will have no effect on the appearance of the exercise when run on a Student Workstation. When WinCALIS runs a script on the Student Workstation it will use the default font specified in the local “wcfnt.inf” file (Times New Roman for ANSI), unless the script instructs it to use some other font.

If you want WinCALIS to use a different font for your lesson, no matter where it is run, you can insert an “L” Default Font Statement at the top of your script (use the Pass Through Window), e.g.,

LArial

Or you can change the font that WinCALIS uses on a window by window basis by inserting a <SET FONT facename> command in that window at the place the font is to take effect. WinCALIS then uses the new font in the window from then on, or until you choose a different font. WinCALIS Author will do this for you semi-automatically via the Fonts menu item in the Scripts Menu.

See Rearranging the Language Menus for how to change the list of fonts displayed in the Fonts dialog box.

(For how to change the ANSI font size, you will need to edit LATIN1.INI. See The Main WinCALIS Initialization File WINCALIS.INI.

Text CALIS authors with lots of text CALIS scripts using pseudo-graphics boxes designed for use with a fixed-width (monospaced) font might prefer to edit in Author using a fixed-width font at “Fine” size. This is more or less possible by editing WCFNT.INF.

Sample Portion of First Part of WCFNT.INF File

```
[FaceNames]
00=Times New Roman
01=Times New Roman
02=Times New Roman
03=Athens
04=Kiev
05=Tel Aviv
06=Sahara
```

The latter part of the WCFNT.INF file contains “magnification” percentages for the fonts used for each character set. E.g., the “Athens” font for Greek is by default magnified to 110% of its basic size.

Sample Portion of Second Part of WCFNT.INF File

```
[03]
Athens=110
```

```
[04]
Kiev = 120
Sofia = 120
```

```
[05]
Tel Aviv = 120
```

```
[06]
Baghdad = 200
Sahara = 200
```

New Configuration Dialog Box Coming Down the Pike:

Author Workstation WinCALIS Startup:

- o Start Author Workstation WinCALIS in Text Conversion (*.CAL) mode
- o Start Author Workstation WinCALIS in No Conversion (*.WCL) mode

Location of Files:

Main Scripts Directory: c:\WinCALIS\scripts
Shared Graphics Directory: c:\WinCALIS\graphics
Shared Multimedia Directory: c:\WinCALIS\mltmedia
Conversion Utilities Directory: c:\WinCALIS\convert
Default Documents Directory: c:\WinCALIS\documents

Keyboard Arrangement (for display in keyboard help):

- o Backslash key (\) on top row, next to Backspace key
- o Backslash key (\) on 2nd row from top, below Backspace key
- o Backslash key (\) on bottom row at extreme right, below Enter key
- o Backslash key (\) on bottom row at extreme left

L:[Language] Menu Top Language: English/W.Europe
(change by setting Default Language Keyboard)

Default Language Keyboard: LATIN1.INI

Author Fonts:

Author ANSI Font: Times New Roman
(change by setting WCFNT.INF 00= line below)

Author Font Size Pixel Height: 21
(change by setting Font Size Medium= below)

LATIN1.INI current settings:

Font Size Pixel Heights:

Tiny=12

Fine=16

Small=19

Medium=21

Large=32

ExtraLarge=48

Huge=66

Fonts:

Font1=Times New Roman

Font2=Arial

Font3=Courier New

Font4=Fixedsys

Font5=MS Sans Serif

Font6=

WCFNT.INF current settings:

[FaceNames]

00=Times New Roman

01=Times New Roman

02=Times New Roman

03=Athens

04=Kiev

05=Tel Aviv

06=Sahara

.

.

Defining Language Keyboard Files

WinCALIS and the WinCALIS Author allow for the remapping of keyboard layouts and the display of non-Roman character sets. The user may select a keyboard mapping and a character set from the pull-down “L:[language]” menu.

Once a language is selected, choosing “keyboard” from the “Help” menu will display the layout of the currently selected font. Accents and key combinations for producing characters are also displayed.

In order to make a character set and a keyboard layout available to the user, three files are required.

1. ***LANGUAGE_NAME.INI***
2. ***LANGUAGE_NAME.WCK***

3. The appropriate character font, contained in font files such as **FONTFILE.FON**

Font Files

Font files contain character sets in a format which can be displayed in Windows.

They should be placed in the subdirectory \FONTS located below the Windows 95 directory.

WinCALIS and WinCALIS Author come bundled with all the fonts necessary to display all character sets.

Language Keyboard “.INI” Files

There is a file ending with the extender .INI for each WinCALIS language keyboard. The .INI file is the initialization file for a particular language keyboard, which specifies the keyboard arrangement, name, fonts, on-line dictionary, and other attributes of the language keyboard.

It must be placed in the WinCALIS home directory--by default \WINCALIS.

You may edit .INI files with a plain text editor such as the Windows Notepad or the DOS EDIT program. For example, you may wish to specify a different on-line dictionary or a new font available for a keyboard. Be sure you understand the particular structure documented below.

The following file “RUSSIANS.INI” is an example of such a file
Comments on the file are preceded with a semicolon (;)

[Language]

Name=Russian-Standard

;The name of the language keyboard that
;appears in the “L:[Language]” pull-down menu.

SubmenuID=8

;This value determines which submenu, if any, the
;keyboard appears in, in this case “Cyrillic”

Face=WC2Uni04

;Not currently used.

RightLeft=0

;The zero value for RightLeft indicate the
;character set is written from left-to-right.
;A value of “1” means right-to-left.

CharSet=0

;This value is either zero, 2, or 255 - ANSI, Symbol or OEM
;(Not currently used)

Keyboard=RussianS

;This is the DOS filename of the .WCK file
;which specifies the keyboard arrangement.

LanguageID=790

;This number must correspond with the ID number
;specified for the RussianS keyboard in the WINCALIS.INI
;main initialization file. The specific number assigned
;will determine the order of the name in the
;L[language]: menu
;The highest possible value for this number is 999.
;See Rearranging the Language Menus

Pitch=2

;This value indicates a proportional font.
;Zero would indicate a fixed font. (Not currently used.)
DeadKey=
;(Not currently used--overstrike (dead) key is always F2
Font1=Kiev
Font2=Moscow
Font3=Sofia
;Typefaces available for use with this keyboard and character
;set are specified here. They will appear as choices when the
;user selects the Set Font menu option in the WinCALIS
;Author Format menu.

[FontSize]
Tiny=12
Fine=16
Small=19
Medium=21
Large=32
ExtraLarge=48
Huge=66

;The adjectives on the left are the names given
;to font point sizes in WinCALIS. The numeric values
;on the right indicate the pixel height for each size.

[Dictionary]
DictFile=wcruseng.dic
IndexFile=wcruseng.idx

;These lines specify the filenames of the on-line dictionary
;and its index file to be accessed in the directory defined in the
;Dictionaries= line of WINCALIS.INI when the user presses F5 or selects
;the menu option Tools, Look Up Word while "Russian-Standard"
;is the current language keyboard. If no filenames are specified,
;the default dictionary with the filenames WINCALIS.DIC and
;WINCALIS.IDX will be used instead.
;See How to Create Your Own On-line Dictionary
;and Setting the Dictionaries Directory

".WCK" Files

(NOTE: Some of the information below is now obsolete.)

This is a binary format file which specifies which keystrokes produce which characters.

It cannot be edited directly. To create a *.WCK file you must edit a text file with the same name and the extension *.MAP. After this file is edited it is processed into the binary file format of a *.WCK file.

After editing such a file select the "Create Keyboard" option from the "File" pull-down menu in WinCALIS Author. The "*.MAP" file will be compiled into a "*.WCK" file.

While the ".MAP" file is not needed to map characters once the ".WCK" file is created, you should save it in case you wish to make changes to the keyboard mapping.

The map file consist of several headings which are written inside of square brackets. ([])

Below each of these topics are the actual characters OR their numeric character values, enclosed in curly braces ({ }), affected by the headings under which they are listed.

Comments can be included in a “.MAP” file by preceding them with a semicolon (;).

Text which appears on a line after a semicolon will be ignored.

If you wish to remap a square brace ([,]), a curly brace ({ , }), or a semicolon (;) you must use the character values for these characters rather than the characters themselves.

To make a “.MAP” file as readable as possible it is recommended that you list each combination of characters on a single line. This is not required.

A simple debugging feature in the program which compiles your “.MAP” file into a “.WCK” file will give an error message telling you which line of the “.MAP” file caused a problem.

Because creating a new “.WCK” file overwrites the previous one (unless you give it a new name) it is recommended that you keep backups.

Also, it is a good idea to develop “.MAP” files incrementally, and convert them to “.WCK” files regularly. This process takes only a few seconds and will make pinpointing errors a simple task.

Note that if you use the WinCALIS Author you can type in character from foreign character sets instead of their character values.

This is not recommended because such files appear unintelligible when other fonts are used to display them or when they are edited with DOS-based word processors.

At the end of the conversion process a message box will appear to confirm that the “.WCK” file has been successfully generated.

Click on one of the following topics for more information on the contents of a “.MAP” file.

Actual “.MAP” file

All of the headings listed below are required in every “.MAP” file, even if they are not used, with the exception of [Position]

The [MAP] heading

The [ALPHA] heading

The [PUNCTUATION] heading

The [SPACE] heading

The [DIGIT] heading

The [VOWEL] heading

The [UPPER/LOWER] heading

The [ACCENT/NOACCENT] heading

The [POSITION] heading

The [ACCENTS] heading

The [ENDV1] heading

Sample “.MAP” File HEBREW1.MAP

;hebrewi.map - Israeli keyboard

```
[map]
t {05D0} ;alef
c {05D1} ;bet
d {05D2} ;gimel
s {05D3} ;dalet
v {05D4} ;he
u {05D5} ;vav
z {05D6} ;zayin
j {05D7} ;het
y {05D8} ;tet
h {05D9} ;yod
l {05DA} ;final kaf
f {05DB} ;kaf
k {05DC} ;lamed
o {05DD} ;final mem
n {05DE} ;mem
i {05DF} ;final nun
b {05E0} ;nun
x {05E1} ;samekh
g {05E2} ;ayin
{003B} {05E3} ;final pe ;mapped to semicolon
p {05E4} ;pe
. {05E5} ;final tsadi
m {05E6} ;tsadi
e {05E7} ;qof
r {05E8} ;resh
a {05E9} ;shin
, {05EA} ;tav
: {05C3} ; Hebrew punctuation sof pasuq
/ .
q /
w ,
S {05B0} ;HEBREW POINT SHEVA
O {05B1} ;HEBREW POINT HATAF SEGOL
I {05B2} ;HEBREW POINT HATAF PATAH
B {05B3} ;HEBREW POINT HATAF QAMATS
A {05B4} ;HEBREW POINT HIRIQ
F {05B5} ;HEBREW POINT TSERE
C {05B6} ;HEBREW POINT SEGOL
H {05B7} ;HEBREW POINT PATAH
D {05B8} ;HEBREW POINT QAMATS
E {05B9} ;HEBREW POINT HOLAM
G {05BB} ;HEBREW POINT QUBUTS
J {05BD} ;HEBREW POINT METEG
K {05BE} ;HEBREW PUNCTUATION MAQAF
L {05BF} ;HEBREW POINT RAPE
M {05C0} ;HEBREW PUNCTUATION PASEQ;;HEBREW POINT PAS
N {FFFF}
P {FFFF}
```

Q {FFFF}
R {FFFF}
T {FFFF}
U {FFFF}
V {FFFF}
W {FFFF}
X {FFFF}
Y {FFFF}
Z {FFFF}

[alpha]

;Unicode 1.1; WinCALIS 1.0 draft Hebrew; Hebrew Text CALIS codes; description

{05B0} ;HEBREW POINT SHEVA
{05B1} ;HEBREW POINT HATAF SEGOL
{05B2} ;HEBREW POINT HATAF PATAH
{05B3} ;HEBREW POINT HATAF QAMATS
{05B4} ;HEBREW POINT HIRIQ
{05B5} ;HEBREW POINT TSERE
{05B6} ;HEBREW POINT SEGOL
{05B7} ;HEBREW POINT PATAH
{05B8} ;HEBREW POINT QAMATS
{05B9} ;HEBREW POINT HOLAM
{05BB} ;HEBREW POINT QUBUTS
{05BC} ;HEBREW POINT DAGESH OR MAPIQ;;HEBREW POINT DAGESH
{05BD} ;HEBREW POINT METEG
{05BE} ;HEBREW PUNCTUATION MAQAF
{05BF} ;HEBREW POINT RAFE
{05C0} ;HEBREW PUNCTUATION PASEQ;;HEBREW POINT PAS
{05C1} ;HEBREW POINT SHIN DOT
{05C2} ;HEBREW POINT SIN DOT
{05C3} ;HEBREW PUNCTUATION SOF PASUQ
{05D0} ;alef
{05D1} ;bet
{05D2} ;gimel
{05D3} ;dalet
{05D4} ;he
{05D5} ;vav
{05D6} ;zayin
{05D7} ;het
{05D8} ;tet
{05D9} ;yod
{05DA} ;final kaf
{05DB} ;kaf
{05DC} ;lamed
{05DD} ;final mem
{05DE} ;mem
{05DF} ;final nun
{05E0} ;nun
{05E1} ;samekh
{05E2} ;ayin
{05E3} ;final pe
{05E4} ;pe
{05E5} ;final tsadi
{05E6} ;tsadi
{05E7} ;qof

{05E8} ;resh
 {05E9} ;shin
 {05EA} ;tav
 {05F0} ;HEBREW LIGATURE YIDDISH DOUBLE VAV
 {05F1} ;HEBREW LIGATURE YIDDISH VAV YOD
 {05F2} ;HEBREW LIGATURE YIDDISH DOUBLE YOD
 {05F3} ;HEBREW PUNCTUATION GERESH
 {05F4} ;HEBREW PUNCTUATION GERSHAYIM
 {FB1E} ;HEBREW POINT JUDEO-SPANISH VARİKA;HEBREW POINT VARİKA;
 {FB1F} ;HEBREW LIGATURE YIDDISH YOD YOD PATAH; [05D9] & [05D9] & [05B7]
 {FB20} ;HEBREW LETTER ALTERNATIVE AYIN;[05E2]
 {FB21} ;HEBREW LETTER WIDE ALEF;[05D0]
 {FB22} ;HEBREW LETTER WIDE DALET;[05D3]
 {FB23} ;HEBREW LETTER WIDE HE;[05D4]
 {FB24} ;HEBREW LETTER WIDE KAF;[05DB]
 {FB25} ;HEBREW LETTER WIDE LAMED;[05DC]
 {FB26} ;HEBREW LETTER WIDE FINAL MEM;[05DD]
 {FB27} ;HEBREW LETTER WIDE RESH;[05E8]
 {FB28} ;HEBREW LETTER WIDE TAV;[05EA]
 {FB29} ;HEBREW LETTER ALTERNATIVE PLUS SIGN;[002B]
 {05CA} ;HEBREW LETTER SHIN WITH SHIN DOT;[05E9] & [05C1]
 {05CB} ;HEBREW LETTER SHIN WITH SIN DOT;[05E9] & [05C2]
 {FB2C} ;HEBREW LETTER SHIN WITH DAGESH AND SHIN DOT;[05E9] & [05BC] & [05C1]
 {FB2D} ;HEBREW LETTER SHIN WITH DAGESH AND SIN DOT;[05E9] & [05BC] & [05C2]
 {FB2E} ;HEBREW LETTER ALEF WITH PATAH;[05D0] & [05B7]
 {FB2F} ;HEBREW LETTER ALEF WITH QAMATS;[05D0] & [05B8]
 {FB30} ;HEBREW LETTER ALEF WITH MAPIQ;[05D0] & [05BC]
 {05CC} ;HEBREW LETTER BET WITH DAGESH;[05D1] & [05BC]
 {FB32} ;HEBREW LETTER GIMEL WITH DAGESH;[05D2] & [05BC]
 {FB33} ;HEBREW LETTER DALET WITH DAGESH;[05D3] & [05BC]
 {FB34} ;HEBREW LETTER HE WITH MAPIQ;[05D4] & [05BC]
 {FB35} ;HEBREW LETTER VAV WITH DAGESH;[05D5] & [05BC]
 {FB36} ;HEBREW LETTER ZAYIN WITH DAGESH;[05D6] & [05BC]
 {FB38} ;HEBREW LETTER TET WITH DAGESH;[05D8] & [05BC]
 {FB39} ;HEBREW LETTER YOD WITH DAGESH;[05D9] & [05BC]
 {FB3A} ;HEBREW LETTER FINAL KAF WITH DAGESH;[05DA] & [05BC]
 {05CF} ;HEBREW LETTER KAF WITH DAGESH;[05DB] & [05BC]
 {FB3C} ;HEBREW LETTER LAMED WITH DAGESH;[05DC] & [05BC]
 {FB3E} ;HEBREW LETTER MEM WITH DAGESH;[05DE] & [05BC]
 {FB40} ;HEBREW LETTER NUN WITH DAGESH;[05E0] & [05BC]
 {FB41} ;HEBREW LETTER SAMEKH WITH DAGESH;[05E1] & [05BC]
 {FB43} ;HEBREW LETTER FINAL PE WITH DAGESH;[05E3] & [05BC]
 {05CD} ;HEBREW LETTER PE WITH DAGESH;[05E4] & [05BC]
 {FB46} ;HEBREW LETTER TSADI WITH DAGESH;[05E6] & [05BC]
 {FB47} ;HEBREW LETTER QOF WITH DAGESH;[05E7] & [05BC]
 {FB48} ;HEBREW LETTER RESH WITH DAGESH;[05E8] & [05BC]
 {FB49} ;HEBREW LETTER SHIN WITH DAGESH;[05E9] & [05BC]
 {05CE} ;HEBREW LETTER TAV WITH DAGESH;[05EA] & [05BC]
 {FB4B} ;HEBREW LETTER VAV WITH HOLAM;[05D5] & [05B9]
 {FB4C} ;HEBREW LETTER BET WITH RAPE;[05D1] & [05BF]
 {FB4D} ;HEBREW LETTER KAF WITH RAPE;[05DB] & [05BF]
 {FB4E} ;HEBREW LETTER PE WITH RAPE;[05E4] & [05BF]
 {FB4F} ;HEBREW LIGATURE ALEF LAMED;[05D0] & [05DC]

[punctuation]

!@#\$%^&*()_+!="<>?

{003B} {005B} {005D} {007B} {007D} ;i.e. ;[] {}

{05C3} ;Hebrew punctuation sof pasuq

[space]

{0020} ; {32}

[digit]

1234567890

[vowel]

[upper/lower]

[accent/noaccent]

{FB2A} {05E9} ;shin shindot

{FB2B} {05E9} ;shin sindot

{FB31} {05D1} ;bet dagesh

{FB44} {05E4} ;pe dagesh

{FB4A} {05EA} ;tav dagesh

[Position]

;isolate, initial, medial, final;

{05D0} {05D0} {05D0} {05D0} ;alef

{05D1} {05D1} {05D1} {05D1} ;bet

{05D2} {05D2} {05D2} {05D2} ;gimel

{05D3} {05D3} {05D3} {05D3} ;dalet

{05D4} {05D4} {05D4} {05D4} ;heh

{05D5} {05D5} {05D5} {05D5} ;vav

{05D6} {05D6} {05D6} {05D6} ;zayin

{05D7} {05D7} {05D7} {05D7} ;het

{05D8} {05D8} {05D8} {05D8} ;tet

{05D9} {05D9} {05D9} {05D9} ;yud

{05DB} {05DB} {05DB} {05DA} ;kaf

{05DC} {05DC} {05DC} {05DC} ;lamed

{05DE} {05DE} {05DE} {05DD} ;mem

{05E0} {05E0} {05E0} {05DF} ;nun

{05E1} {05E1} {05E1} {05E1} ;sameh

{05E2} {05E2} {05E2} {05E2} ;ayin

{05E4} {05E4} {05E4} {05E3} ;peh

{05E6} {05E6} {05E6} {05E5} ;tsadi

{05E7} {05E7} {05E7} {05E7} ;qof

{05E8} {05E8} {05E8} {05E8} ;resh

{05E9} {05E9} {05E9} {05E9} ;shin

{05EA} {05EA} {05EA} {05EA} ;tav

[accents]

shindot ' ; {05C1}

sindot ` ; {05C2}

dagesh , ; {05BC}

[shindot]

{05E9} {FB2A} ;shin

[sindot]

{05E9} {FB2B} ;shin

```
[dagesh]
{05D1} {FB31} ;bet
{05E4} {FB44} ;pe
{05EA} {FB4A} ;tav

[endv1]
```

“.MAP” File Heading [MAPPING]

This heading is used to map or remap characters to the keyboard.

Mappings are listed in pairs.

First, the keyboard character which is to be replaced is listed.

Second, the character which will replace it is listed.

For example

```
[MAPPING]
```

```
a f
```

```
{0064} {009B}
```

As specified above, pressing the “a” key on the PC keyboard will no longer produce an “a” but rather an “f” instead.

In the same way, pressing the “d” key (character value 0064) will produce character 009B instead.

You may specify characters by entering them directly or by their character values enclosed in curly braces.

Separate each entry with a least one blank space.

Note that the first value or character must be one of the keys located on the PC keyboard.

Depending on what font is used to display these files, you may see Cyrillic, Greek, or Roman characters, among others.

Files created using a keyboard mapping intended for Russian will look like garbage if they are displayed with another font.

When users select the keyboard help feature in UniEdit, the keyboard window will display the characters as you have remapped them.

“.MAP” File Heading [ALPHA]

This heading is used to list all of the characters in a font which are alphabetic.

This feature is useful because it allows the script author to write diagnostic messages based on the presence or absence of an alphabetic character.

Characters may be listed under this heading by entering them directly or by typing in their character value enclosed in curly braces.

“.MAP” File Heading [PUNCTUATION]

This heading is used to specify all of the characters in a font which are used for punctuation.

WinCALIS uses this information to determine if a student response is correct or incorrect based on the improper use of punctuation.

Characters under this heading may be entered directly or by listing their character value in curly braces.

“.MAP” File Heading [SPACE]

Specify what character (or possibly characters) is used as the space character.

WinCALIS uses this information to ignore multiple spaces between words in a student response.

This character must be listed by its character value. (You can’t display an empty character!)

This character is nearly always {0020}.

“.MAP” File Heading [DIGIT]

Specify which characters are numeric.

At this time this feature is not used. Potentially it could be used to distinguish answers which contain numeric characters from other answers. Call us if you need this for your courseware development.

Although unused, every “.MAP” file must contain this heading. No characters need follow it.

Characters can be entered directly or by listing their character value inside of curly braces.

“.MAP” File Heading [VOWEL]

This heading is only useful for Chinese at present. If you feel that the ability of WinCALIS to distinguish consonants and vowels would be useful for some other language, this feature could easily be added.

This heading must appear in all “.MAP” files.

Characters may be entered directly or by listing their character value inside of curly braces.

“.MAP” File Heading [UPPER/LOWER]

The information in this and the following section is no longer used by WinCALIS/UniEdit. The global information file WINCALIS.WGI (q.v.) is used instead.

This heading is used to list uppercase/lowercase equivalents.

This information is used by WinCALIS to determine whether case is significant in judging student responses.

Characters under this heading are listed in pairs.

First the upper case character or its character value in curly braces.

Second the lower case character or its character value in curly braces.

“.MAP” File Heading [ACCENT/NOACCENT]

This heading is used to list accented/unaccented equivalents.

This information is used by WinCALIS to determine whether the proper use of accents is significant in judging student responses.

Characters under this heading are listed in pairs.

First the accented character or its character value in curly braces.

Second the unaccented character or its character value in curly braces.

“.MAP” File Heading [POSITION]

This is the only optional heading in a “.MAP” file.

It is used to specify what characters are affected by their position in a word.

Characters are listed in sets of four in the following order:

{[independent]} {[initial]} {[medial]} {[final]}

Characters may be listed by entering the characters themselves or by listing their character values in curly braces.

This feature is used with the UniEdit Persian and Arabic fonts.

“.MAP” File Heading [ACCENTS]

This heading is used to precede a listing of all of the names given to diacritic marks.

After the name of the diacritic, list the single character which will be used in combination with the overstrike key F2 to form an accented character.

Diacritic names must not contain spaces.

For example:

[ACCENTS]

umlaut "

nonsense {0078}

Two accent keys are now defined with the names “umlaut” and “nonsense.”

When keyboard help is selected from within UniEdit, a keyboard will display with the double quote (") and character 0078 - a lowercase “x” in pale blue.

This indicates to the user that these keys can be used in combination with the dead key to produce a new character.

Beneath the [ACCENTS] heading each accent is listed a second time as a heading itself.

Following this is a listing of all characters which can be typed and then altered or changed completely with the overstrike key/accent key combination.

For example:

```
[umlaut]
a {00E4}
{0075} {00FC}
[nonsense]
a z
{0064} b
```

As specified above, the key combination “a” + the dead key + the “umlaut” accent key (a double quote as defined in the [accents] section) will produce character 00E4 - an “a” with two dots over it.

In the same way, a lower case “u” (character 0075) becomes a “u” with two dots over it (character 00FC).

The “nonsense” accent key changes a lower case “a” into a lower case “z.”

Character 0064 (a lower case “d”) becomes a lower case “b.”

With the keyboard help window open the user can click with the secondary mouse button on an accent key to see which keys it affects and what characters they produce. The selected accent key turns red and the affected keys display in green.

According to the examples above, if the user opens the keyboard window and clicks with the secondary mouse button the “nonsense” accent key the lower case “x” turns from pale blue to green. The two characters it affects - the “a” and the “d” are redisplayed as a “z” and a “b.”

Clicking on or pressing the “umlaut” accent key (the double quote) changes the “a” and the “u” to umlauted characters.

All listings under accent headings must be paired - original character and character to be displayed after the overstrike key/accent key combination is struck.

Characters may be entered directly or listed by character value within curly braces.

“.MAP” File Heading [ENDV1]

No characters are listed beneath this heading. It simply marks the end of the file. However, there should be at least one new line after the heading.

The WinCALIS Global Information File WINCALIS.WGI

The binary global information file WINCALIS.WGI in WinCALIS 2.x replaces much of the language-specific information which was formerly in the individual language keyboard *.WCK files WinCALIS 1x regarding what constitutes punctuation, capitalization (upper-and-lower case relationships), and accented-unaccented character relationships. WinCALIS now is able to evaluate these linguistic and orthographic characteristics properly, no matter which keyboard is currently selected.

Note that the [upper/lower] section of WINCALIS.WGI is used for case-independent searches in the Find and Replace dialog boxes. It could also be used in the future for sorting purposes, along with the [accent/noaccent] section.

The name WINCALIS.WGI is arbitrary and is identified to the WinCALIS system by the “GlobalInfoFile=” line in WINCALIS.INI. By default, the file is called WINCALIS.WGI and is in the main program directory.

WINCALIS.WGI is created from its ASCII text counterpart WINCALIS.INF by opening WINCALIS.INF in the Scratchpad as an ASCII/ANSI file, then clicking on the Create Global File menu item in the File menu. WINCALIS.INF is included in the WinCALIS Author Workstation, for browsing and editing on the part of courseware authors. Please notify the Humanities Computing Laboratory if you find errors or omissions in the file.

Samples entries under each subheading in WINCALIS.WGI

```
[non-spacingmarks] (unused)
[punctuation]
{0021} ;!
{0022} ;"
```

```
{0023} ;#
{0024} ;$
.
.
{037E} ;Greek question mark (semicolon)
{05C3} ;Hebrew
{060C} ;Arabic comma
{061B} ;Arabic semicolon
{061F} ;Arabic question mark
{066A} ;Arabic percent sign
{06D4} ;Arabic period (Farsi period) (Urdu)?
{1255} ;Ethiopian (Amharic) wordspace
{1256} ;Ethiopian (Amharic) comma
.
.
[upper/lower]
A a
B b
C c
D d
E e
.
.
{00DC}{00FC} ;U diaeresis u diaeresis
{00DD}{00FD} ;Y acute y acute
{00DE}{00FE} ;capital thorn small thorn
{00DF}{00DF} ;esszed esszed
{0100}{0101} ;A macron a macron
{0102}{0103} ;A breve a breve
{0104}{0105} ;A ogonek a ogonek
{0106}{0107} ;C acute c acute
{0108}{0109} ;C circumflex c circumflex
.
.
[accent/noaccent]
{00C0} A ;A grave
{00C1} A ;A acute
{00C2} A ;A circumflex
{00C3} A ;A tilde
.
.
{FB44}{05E4} ;pe dagesh
{FB4A}{05EA} ;tav dagesh
{FD15}{00EA} ;e circumflex grave e circumflex
{FD39}{00EA} ;e circumflex acute e circumflex
{FD60}{00EA} ;e circumflex macron e circumflex
{FD61}{00EA} ;e circumflex hacek e circumflex
.
.
[endv1]
```

Deleting Nonessential Files to Save on Disk Space

When you install WinCALIS, it installs *.INI, *.MAP, and *.WCK files for all available languages into your program directory by default. The Delete Language menu item in the L:[Language] menu deletes languages from the listing in the menu, but it does not delete any files from your disk.

If you need to economize on disk space, after deleting a language from your menu, you may also use the Windows File Manager, DOS commands, or another tool to delete its *.INI, *.MAP, and *.WCK files from your hard disk, or move them to a backup storage location.

*.MAP files are simply the text versions of binary keyboard *.WCK files, which are those used by WinCALIS. If you do not anticipate the need to edit or add any keyboard file, you may delete *all* the *.MAP files from your system. See the section below for other nonessential files you may delete to conserve disk space.

If necessary, you can always recover deleted files by reinstalling the Author Workstation.

Installing MCI Drivers For Multimedia

Installation in Control Panel

Disk 4 of the WinCALIS Author Workstation Setup and Disk 2 of the WinCALIS Student Workstation Setup includes some runtime drivers for Windows 3.1 MCI (Media Control Interface). These drivers variously allow an author or student workstation to display previously-created animation files, or to control an MCI-compatible laserdisc player or VCR. The drivers can be installed using the following procedure:

- 1) Load the Windows 3.1 Control Panel (from Main group).
- 2) Double-click on the Drivers applet.
- 3) Select the Add button.
- 4) Highlight the "Unlisted or Updated Driver" item from the List of Drivers and click on OK.
- 5) Insert the WinCALIS Setup diskette labeled "Disk 4: Multimedia, MCI Drivers" (or Disk 2) in drive A or other drive, and enter the path where these files can be found, a:\drivers, and click OK.
- 6) Select the desired driver.
- 7) Repeat for as many drivers as needed.

Making MCI Drivers Work in WinCALIS

In order that WinCALIS, WinCALIS Multimedia Editor, and the Windows Media Player be able to recognize the media device needed to play a file from the file extension, in the case of "compound" devices (i.e., files) such as *.WAV digital sound files, *.AVI Video for Windows files, *.MMM Microsoft Multimedia Movie Player animation files, etc., the following extra step is also needed:

Make a backup copy of your main Windows initialization file, "WIN.INI" in your main Windows program directory, using the Windows File Manager or DOS "COPY" command, e.g. as "WININI.BAK". Then using the Windows Notepad, the DOS editor, or other plain text editor, open the

file "WIN.INI", find the [mci extensions] section, and add as many as you like of the following lines, if they do not already appear there.

```
wav=waveaudio
mid=sequencer
rmi=sequencer
avi=AVIVideo
mmm=MMMovie
fli=Autodesk
flc=Autodesk
awm=Animation
awa=Animation
mov=QTWVideo
```

Other MCI Information

The word following the = sign must correspond to a line in the [mci] section of your Windows SYSTEM.INI file, which identifies the MCI driver for that device, of the form "MMMovie=mcimmp.drv". Such a line should have been written by the Windows Control Panel for each MCI driver which you chose to install in steps 1-7 above. The MCI drivers themselves, along with any supporting *.DLL files, etc., are usually added to your Windows system directory.

Installing WinCALIS on a Network

Note: If you are installing WinCALIS on a workstation which is running a network copy of Windows, you must have the necessary network privileges to write to the network Windows system directory, even if WinCALIS, is being installed on a local drive, since WinCALIS must install its fonts and a few other system files in the common Windows directory.

WinCALIS performs well in a network setting. But there are certain aspects of WinCALIS performance which you should be aware of.

WinCALIS Home Directory

It is important to be aware of the drive letter for the "Home" and other directories specified in WINCALIS.INI. See the various sections on changing subdirectory defaults in The Main WinCALIS Initialization File WINCALIS.INI. It is sometimes the case with installation on a network server that the DOS drive letter by which that server is known to a local workstation changes. If that happens, WinCALIS will not function properly on the local workstation. You will need to bring the drive letters specified in WINCALIS.INI into line with the expectations of the local workstations.

Writing *.WCL Files

Another aspect of network behavior is that when *.CAL script files are executed by WinCALIS, it must convert them to *.WCL files, and similarly convert text versions of any ancillary files to their binary equivalents. So WinCALIS must have write privileges in the directories where such files are stored. This does not apply to the Student Workstation version of WinCALIS, which is not capable of converting *.CAL files and does not write any files to disk.

Double dots ".."

The drives/directory window in the WinCALIS opening screen uses a Windows 3.0-style interface, which requires clicking on the "double dots" in order to be able to move up to the parent directory. On

a Novell network these dots are not automatically displayed as they are in DOS. To display the dots in WinCALIS running on a Novell server, be sure that the line

show dots = on

appears in the SHELL.CFG file for every workstation which will run WinCALIS from the network.

Non-ASCII Characters in DOS filenames

Windows interfaces with the Novell operating system in slightly different ways from the ways it interfaces with DOS. One of these is the way it translates non-ASCII characters, such as accented characters, back and forth between the DOS/IBM-PC extended ASCII character codes and the ANSI (Latin1 or ISO 8859)) character codes used within Windows.

Tips on Installing WinCALIS on a Shared Network Drive from a Local Workstation

See the note at the beginning of this topic on necessary network privileges.

Install WinCALIS in the desired directory on the shared network drive from one workstation. This will create a complete installation in that workstation's copy of Windows, including the Program Manager group with icons and the fonts installed by WinCALIS Setup.

Then in order to give WinCALIS icons for the copies of Windows running on the other workstations, you will need to copy the group file WINCALIS.GRP from that workstation's Windows directory to every other workstation's Windows directory, and add the following line to each progman.ini file:

```
Group8=C:=\WINDOWS\WINCALIS.GRP
```

(or whatever group number has been created in the original PROGMAN.INI).

You will also need to install the WinCALIS fonts which were installed in the Windows System directory (e.g., \windows\system) of the workstation used for the initial Setup (assuming that it is running a local copy of Windows) into the copies of Windows running on the other workstations. Copy all the files wc2*.fon and wc*.ttf into the other windows\system directories.

Then run Control Panel in the Main group, then the Fonts applet, then Add, and change to the \windows\system directory. After logging the fonts in that directory, holding down Ctrl, click on all fonts beginning with "WinCALIS 2.0...". If you alternatively copy the lines from the original machine's win.ini file into the other machines' win.ini files, you will also need to copy the respective "wc*.fot" files from the original machine.

There are two other required files to copy into the Windows System directories:

```
ctl3d.dll
pxengwin.dll
```

If the subdirectory and Windows desktop arrangements are the same on all the workstations, you can make a little batch file to do these copying jobs effortlessly. Put all the above files and the batch file, e.g., "TRANSFER.BAT," in a temporary location somewhere on the network, say, the Z: drive. Then run that batch file from each workstation. Your batch file might look like this:

```
copy z:\WinCALIS.grp c:\windows
copy c:\windows\progman.ini c:\windows\progmanb.ini
copy z:\progman.ini c:\windows
copy z\wc2*.fon c:\windows\system
```

```
copy z:wc*.ttf c:\windows\system
copy z:ctl3d.dll c:\windows\system
copy z:pxengwin.dll c:\windows\system
```

From each workstation, you would type “z:transfer” to perform the needed transfer of files.

How to Uninstall WinCALIS and WinCALIS Author

To uninstall WinCALIS 2.x you will need to remove the directories and files listed below from your machine. You can use the Windows File Manager, or you can use DOS commands. WinCALIS 2.x also adds fonts to your Windows system, causing Windows to add them to the [fonts] section of your WIN.INI file. WinCALIS makes no other modifications directly to any of your Windows initialization files, such as WIN.INI, SYSTEM.INI, etc.

Uninstalling Using the Windows File Manager:

- 1) Click on the File Manager icon in the Windows 3.1 Main group to start the File Manager.
- 2) Select the drive which contains the directory which you specified as the WinCALIS program directory during setup, e.g., the C: drive.
- 3) Highlight the WinCALIS program directory (by default C:\WINCALIS). Then press the Delete key. When asked, press “Delete All”.

WARNING!! This will delete ALL files in ALL subdirectories under the WinCALIS program directory--over 600 files in the case of a full installation. Be sure to exercise caution that you have specified the correct directory, and make backup copies first of any of your own script or document files which you wish to keep.

- 4) Still using the File Manager, locate the main Windows directory (e.g., C:\WINDOWS) and click on it in order to display the files in it. Delete the following file by highlighting it and pressing the Delete key:

MLPTT.INI

- 5) Now locate the Windows System directory (e.g., C:\WINDOWS\SYSTEM) and click on it in order to display the files in it. Delete the following files:

WARNING!! Other applications may have also installed one or more of these files and still need them. If you are in doubt, do NOT delete them. They will not affect the performance of your system.

PXENGWIN.DLL	Paradox Engine library for record-keeping
CTL3D.DLL	3-D Effects in dialog boxes

If the following files were added through the Control Panel, Drivers, Add Driver procedure described in a separate section below, they should also be removed through the Control Panel.

AAPLAY.DLL	Autodesk Animation Player
------------	---------------------------

MCIAAP.DRV	“
MCIAWI.DRV	Gold Disk Animation
MCIAWIID.DLL	“
MCIAWIIO.DLL	“
MCIAWIIT.TSK	“
MCIMMP.DRV	Microsoft Multimedia Movie Player
MCIPANAS.DRV	Panasonic Videodisc Player
MCIPIONR.DRV	Pioneer Videodisc Player
MCIVISCA.DRV	ViSCA-compatible VCRs
MCIVSONY.DRV	Sony Videodisc Player
MMP.DLL	Microsoft Multimedia Movie Player

6) Now close the Windows File Manager, and click on the icon for the Control Panel, also in the Main group, to open the Windows Control Panel.

7) Click on the Fonts applet.

8) Scroll through the list of “Installed Fonts” until you see the group of fonts described as “WinCALIS 2.0.....”. All such fonts were added only as part of the WinCALIS 2.0 Setup, and may be safely removed as part of the Uninstall process. Highlight all these font names by either of the following procedures:

- a) While holding down the Shift key, click on the first font beginning with “WinCALIS 2.0...”, then scroll to the last font beginning with “WinCALIS 2.0...” and click on it.
- b) While holding down the Ctrl key, click on each font which begins with “WinCALIS 2.0...”.

(This procedure may also be used by any WinCALIS user to selectively remove unwanted fonts from his Windows system, in order to conserve disk space, speed up Windows startup time, etc. For example, a WinCALIS Author who deals only with French, German, or Spanish, and is willing to confine himself to the ANSI/Latin1 character set, may choose to remove all of these extra “WinCALIS 2.0” fonts, or at least any of the fonts for Unicode pages with exotic alphabets.)

9) Click on the Remove button. When asked for confirmation, click in the check box which says “Delete font files from disk” and press the “Yes to All” button to uninstall all fonts installed during the WinCALIS 2.0 Setup.

10) Close the Fonts Applet and the Windows Control Panel.

WinCALIS is now completely uninstalled.

Chapter 11:

WinCALIS Code and Command Conversions

Eight-bit Code Conversions
Sixteen-bit Code Conversions
Twenty-four-bit Code Conversions
UTF-7 and UTF-8 Conversions
Conversion of Commands in Older WinCALIS Scripts

Eight-bit Code Conversions

*Conversion of WinCALIS 1.0, WinCALIS 1.1, Text CALIS *.CAL Script Files and other 8-bit Code Text Files to and from WinCALIS 2.x*

The stand-alone DOS utility WC2&8BIT.EXE may be used to convert single or multiple text files, including *.CAL script files, from 8-bit (single-byte) codes to WinCALIS 2.x 16-bit Unicode (ISO 10646) code points. It can also perform the reverse function of converting WinCALIS 2.x 16-bit Unicodes to 8-bit (single-byte) codes.

Single WinCALIS 1.1 ANSI (Latin1) text files may also be converted upon opening them in a WinCALIS Author document window by selecting “ANSI/WinCALIS 1.1” in the “List Files of Type” box in the File, Open dialog box, then saving them as WinCALIS Author Unicode files (select the appropriate file type in the File, Save “Save File As Type” box). Or they may be converted by copying them to the Windows clipboard from Notepad, Microsoft Write, or other ANSI text editor, then pasting them into WinCALIS Author and saving them as WinCALIS Author Unicode files.

Non-ANSI 8-bit-coded files may be converted upon opening them in WinCALIS Author by selecting “Other conversions” in the File, Open “List Files of Type” box. Or they may similarly be converted by copying them to the Windows clipboard from another application, then using the Paste As... option in the WinCALIS Author Edit menu to convert codes “on the fly” upon pasting them into WinCALIS Author, and saving them as WinCALIS 2.x Unicode files or another Unicode file type.

Screens of DOS-IBM-extended ASCII coded files, such as generic Text CALIS scripts, will be converted to ANSI in the process of copying from a DOS window to the Windows Clipboard, but this procedure is slow for files longer than a few screens.

WC2&8BIT.EXE can be found in the “CONVERT” subdirectory under your WinCALIS Author program directory if you installed the option “Code Conversion Utilities” during Setup. In order to access this utility complete the following steps:

- Go to a DOS prompt and change directories to the “CONVERT” subdirectory.
- At the prompt type:

```
wc2&8bit sourcefile targetfile [/n] [/r] [/t:tablefile]
```

The /n switch ignores all file headers (such as the WinCALIS/Uniedit Unicode file header), in both reading and writing files.

The direction of conversion is by default from an 8-bit-code source file to a UNICODE target file. The /r switch reverses the direction of the conversion, that is, from a WinCALIS Author source file to an 8-bit-code target file.

The /t:tablefile switch causes the specified table file to be used for conversion. If no table name is specified on the command line, WC2&8BIT.EXE will use any table it finds in the current directory with the name WC2&8BIT.TAB. If a table name is specified, WC2&8BIT.EXE will use that table instead.

There is limited support for DOS wild cards and paths, e.g.

WC2&8BIT *.CAL NEW

where 'new' is a subdirectory under the current directory to which converted versions of all files with matching filename *.CAL will be written. Or:

WC2&8BIT *.TXT *.UNI

where all source files with file extension .TXT are converted to target files with file extension .UNI.

Many conversion tables are supplied in the CONVERT directory. It is also possible to compile your own table file. The table should be a binary file with codes 1-255 (no 0) using the following structure:

bytes 1-2 = UNICODE 1 (Little-endian [Intel integer] order)

bytes 3-4 = UNICODE 2

.

.

bytes 509-510 = UNICODE 255.

A *.TAB table with this structure can be produced with the utilities LST2TAB3.EXE or LST2TAB5.EXE, in the CONVERT subdirectory, from a plain text file *.LST, with one code correspondence per line and with the codes on each line separated by tabs. Use LST2TAB3.EXE if the first code is expressed in decimal and the second code (UNICODE) is expressed in hexadecimal, e.g.,

001(tab)0001

.

.

255(tab)00A0.

Use LST2TAB5.EXE if the first code in each line is expressed in hexadecimal, rather than in decimal, e.g.,

01(tab)0001

.

.

FF(tab)00A0.

If you name your *.TAB file WC2&USER.TAB, it will also be available from within WinCALIS Author and UniEdit, via the File,Open, File,Save and Edit,Paste As... options, using the "User-Supplied 8-bit Char. Set" item in the "Convert From" or "Convert To" dialog box.

Key to supplied conversion tables to be used with WC2&8BIT.EXE, with filename of type WC2&???.TAB:

WC2&ANSI.TAB ASCII/ANSI(=ISO 8859/1 "Latin1", ECMA-94)

WC2&D437.TAB ASCII/DOS Code Page 437 (U.S. DOS)/IBM

"Extended ASCII" (Generic Text CALIS)

WC2&D850.TAB	DOS Code Page 850 (Multilingual)
WC2&D852.TAB	DOS Code Page 852 (Latin Slavic)
WC2&D855.TAB	IBM Code Page 855 (Cyrillic)
WC2&D857.TAB	DOS Code Page 857 (Turkish)
WC2&D860.TAB	DOS Code Page 860 (Portugal)
WC2&D861.TAB	DOS Code Page 861 (Icelandic)
WC2&D863.TAB	DOS Code Page 863 (Canadian-French)
WC2&D864.TAB	IBM Code Page 864 (Arabic)
WC2&D865.TAB	DOS Code Page 865 (Nordic)
WC2&D866.TAB	DOS Code Page 866 (Cyrillic)
WC2&D869.TAB	IBM Code Page 869 (Greek)
WC2&PCM4.TAB	PC Code Page M4
WC2&1004.TAB	PC Code Page 1004
WC2&1040.TAB	PC Code Page 1040 (Korean Hangul alphabet)
WC2&1041.TAB	PC Code Page 1041 (Japanese Katakana)
WC2&1043.TAB	PC Code Page 1043 (other East Asian SBCS)
WC2&1250.TAB	Microsoft Windows Code Page 1250 (E.Europe)
WC2&1251.TAB	Microsoft Windows Code Page 1251 (Cyrillic)
WC2&1252.TAB	Microsoft Windows Code Page 1252 (Windows ANSI)
WC2&1253.TAB	Microsoft Windows Code Page 1253 (Greek)
WC2&1254.TAB	Microsoft Windows Code Page 1254 (Turkish)
WC2&1255.TAB	Microsoft Windows Code Page 1255 (Hebrew)
WC2&1256.TAB	Microsoft Windows Code Page 1256 (Arabic)
WC2&1257.TAB	Microsoft Windows Code Page 1257 (Baltic)
WC2&KOI7.TAB	KOI-7 (Cyrillic)
WC2&GO74.TAB	GOST 19768-74 (Old KOI-8 Cyrillic)
WC2&BRAV.TAB	Brjabrin's Alternativnyj Variant (Cyrillic)
WC2&BROV.TAB	Brjabrin's Osnovnoj Variant (Cyrillic)
WC2&RSKR.TAB	AATSEEL "REX" (Cyrillic)
WC2&GERD.TAB	"Gerdsooz" encoding for Persian Farsi
WC2&ISCB.TAB	ISCII 1988 Indian Standard (Bengali)
WC2&ISCD.TAB	ISCII 1988 Indian Standard (Devanagari)
WC2&ISCG.TAB	ISCII 1988 Indian Standard (Gurmukhi)
WC2&ISCJ.TAB	ISCII 1988 Indian Standard (Gujarati)
WC2&ISCK.TAB	ISCII 1988 Indian Standard (Kannada)
WC2&ISCM.TAB	ISCII 1988 Indian Standard (Malayalam)
WC2&ISCO.TAB	ISCII 1988 Indian Standard (Oriya)
WC2&ISCT.TAB	ISCII 1988 Indian Standard (Tamil)
WC2&ISCU.TAB	ISCII 1988 Indian Standard (Telugu)
WC2&ISO1.TAB	ISO 8859/1 ("Latin1") (same as WC2&ANSI.TAB)
WC2&ISO2.TAB	ISO 8859/2 ("Latin2"-Eastern Europe, etc.)
WC2&ISO3.TAB	ISO 8859/3 ("Latin3"-Eastern Europe, etc.)
WC2&ISO4.TAB	ISO 8859/4 ("Latin4"-Eastern Europe, etc.)
WC2&ISO5.TAB	ISO 8859/5 (=ECMA-113,GOST 19768-87, New KOI-8 Cyrillic)
WC2&ISO6.TAB	ISO 8859/6 (=ECMA-114,ASMO 449) (Arabic, Farsi, etc.)
WC2&ISO7.TAB	ISO 8859/7 (=ECMA-118,ELOT 928) (Greek)
WC2&ISO8.TAB	ISO 8859/8 (Hebrew)
WC2&ISO9.TAB	ISO 8859/9
WC2&ISOA.TAB	ISO 6438 (=DIN 31625) (African Extended Latin)
WC2&ISOG.TAB	ISO 5428-1980 (Bibliographic Greek)
WC2&MACA.TAB	Macintosh Arabic Character Set
WC2&MACC.TAB	Macintosh Cyrillic Character Set
WC2&MACE.TAB	Macintosh Central Europe Character Set

WC2&MACG.TAB	Macintosh Greek Character Set
WC2&MACI.TAB	Macintosh Icelandic Character Set
WC2&MACL.TAB	Macintosh Latin (Roman) Character Set
WC2&MACN.TAB	Macintosh Croatian Character Set
WC2&MACR.TAB	Macintosh Romanian Character Set
WC2&MACS.TAB	Macintosh (Adobe) Symbol Character Set
WC2&MACT.TAB	Macintosh Turkish Character Set
WC2&MACU.TAB	Macintosh Ukrainian Character Set
WC2&TIS6.TAB	Thai Industrial Standard 620-2529
WC2&USER.TAB	User-supplied 8-bit Character Set
WC2&VSCI.TAB	VISCII Vietnamese Standard
WC2&WC1B.TAB	WINCALIS 1.0 BASE.FON (English,French, German,Italian,Portuguese,Spanish)
WC2&WC1C.TAB	WINCALIS 1.0 CYRILLIC.FON (Russian)
WC2&WC1E.TAB	WINCALIS 1.0 EEUR.FON (Polish, Hungarian, generic East Europe, etc.)
WC2&WC1G.TAB	WINCALIS 1.0 PGREEK.FON (Greek)
WC2&WC1J.TAB	WINCALIS 1.0 HIRAGANA.FON (Japanese hiragana)
WC2&WC1K.TAB	WINCALIS 1.0 KATAKANA.FON (Japanese katakana)
WC2&WC1V.TAB	WINCALIS 1.0 VIETNAME.FON (Vietnamese)
WC2&CALA.TAB	Amharic Text CALIS
WC2&CALE.TAB	East European Text CALIS
WC2&CALG.TAB	Greek Text CALIS
WC2&CALH.TAB	Hebrew Text CALIS
WC2&CALP.TAB	Portuguese Text CALIS
WC2&CALR.TAB	Russian Text CALIS

The files WC2&D437.LST (and LST2TAB3.EXE) and WC2&ISO1.LST (and LST2TAB5.EXE) can serve as templates from which to create more conversion tables for use with WC2&8BIT.EXE.

For conversion between various 2-byte coding standards (e.g., for Chinese, Japanese, Korean, Unicode 1.0 or 1.1) and WinCALIS Unicode 2.0, use the following separate utilities (requires WinCALIS Author Workstation Setup Disk 9):

- WC2&16BT.EXE for 16-bit fixed-width coded files
- DCT2UNI.EXE for Duke Chinese Typist 2.0 coded files
- WC2&EAST.EXE for JIS, Shift-JIS, EUC, Big5, GB, KS, and other 7-bit modal or mixed 16-bit and 8-bit coded files.
- WC2&24BT.EXE for 24-bit fixed-width coded files such as CCCII or EACC/ANSI Z39.64
- UCS2&UTF.EXE for UTF-7 and UTF-8 coded Unicode files.

See Also: Sixteen-bit Code Conversions
Twenty-four-bit Code Conversions
UTF-7 and UTF-8 Conversions

Sixteen-bit Code Conversions

Conversion of East Asian double-byte files consisting of modal 7-bit or mixed 16-bit and 8-bit codes to and from WinCALIS Author

The stand-alone DOS utility WC2&EAST.EXE may be used to convert single or multiple text files from modal 7-bit or mixed 16-bit/8-bit coding standards such as JIS, SHIFT-JIS, KS, Big5, or GB code to WinCALIS 16-bit Unicode (ISO 10646) files. It can also perform the reverse function of converting

WinCALIS 16-bit Unicodes to modal 7-bit or mixed 16-bit and 8-bit coding standards. For modal 7-bit coded files, use the “EUC” coding table (e.g., for Japanese, use the table EUCJ&UNI.TAB).

Single files can be converted by opening or saving them in WinCALIS Author and selecting “Other conversions” in the “List files of type” or “Save file as type” box. Or they may be converted by copying them to the Windows clipboard from the source application, then pasting them into WinCALIS Author using the Paste As... option in the Edit menu to convert codes “on the fly” during pasting, and saving them as WinCALIS Unicode files.

WC2&EAST.EXE can be found in the CONVERT subdirectory under your WinCALIS program directory. In order to access this utility, do the following:

- Go to a DOS prompt and change to “CONVERT” subdirectory
- At a DOS prompt type the command line:
`wc2&east sourcefile targetfile [/n] [/r] [/m] [/t:tablefile.tab]`

The /n switch ignores all file headers (such as the WinCALIS/Uniedit Unicode file header), in both reading and writing files.

The direction of conversion is from a modal 7-bit or mixed 16-bit/8-bit source file to a WinCALIS Unicode target file by default. The /r switch reverses the direction of the conversion, that is, from a WinCALIS source file to a modal 7-bit or mixed 16-bit/8-bit-code target file.

The /m switch, when used in conjunction with the /r switch, specifies conversion from WinCALIS Unicode source file to a modal 7-bit target file (i.e., a file in which all codes are 7-bit codes, with language shifts marked by Escape codes).

The /t:tablefile switch causes the specified table file to be used for conversion. If no table name is specified on the command line, WC2&EAST.EXE will use any table it finds in the current directory with the name WC2&EAST.TAB. If a table name is specified, WC2&EAST.EXE will use that table instead.

There is limited support for DOS wild cards and paths ,e.g.,

`WC2&EAST *.GB NEW`

where 'new' is a subdirectory under the current directory to which converted versions of all files with matching filename *.GB will be written. Or:

`WC2&EAST *.GB *.UNI`

where all source files with file extension .GB are converted to target files with file extension .UNI.

Several conversion tables are supplied in the CONVERT subdirectory. It is also possible to compile your own table file. The table should be a binary file with the following structure:

bytes 1-2 = Source (GB, Big5, etc.) code 1 (Little-endian byte-reversed)
bytes 3-4 = Unicode 1 (Big-endian)
bytes 5-6 = Source (GB, Big5, etc.) code 2
bytes 7-8 = Unicode 2
.
.

A *.TAB table with this structure can be produced with the utility SJIS2WC2.EXE, in the CONVERT subdirectory, from a plain text file *.LST, with one code correspondence per line and with the codes on each line separated by tabs, e.g.,

8140(tab)3000

.

.

EAA4(tab)7199.

Usage: SJIS2WC2 LISTFILE.LST

Key to supplied conversion tables to be used with WC2&EAST.EXE, (rename as WC2&EAST.TAB or type filename after /t:tablefile switch):

B5&WCUNI.TAB	Big-5 codes (Taiwan Chinese)
EUCJ&UNI.TAB	Extended Unix Codes (Japanese)
(use this table for JIS modal 7-bit conversion)	
GB&WCUNI.TAB	GB-2312-80 codes (simple-style Chinese)
KSJH&UN1.TAB	Korean Johab codes --> Unicode 1.1
KSJH&UN2.TAB	Korean Johab codes --> Unicode 2.0
KSW&UN1.TAB	KS-C5601 Korean Standard Wansung --> Unicode 1.1
KSW&UN2.TAB	KS-C5601 Korean Standard Wansung --> Unicode 2.0
KSUH&UN1.TAB	Korean "Unified Hangul" codes --> Unicode 1.1
KSUH&UN2.TAB	Korean "Unified Hangul" codes --> Unicode 2.0
SJIS&UNI.TAB	Shift-JIS codes (Japanese)

The corresponding files B5&WCUNI.LST, EUCJ&UNI.LST, GB&WCUNI.LST, KSJH&UN1.LST, etc. can serve as templates from which to create more conversion tables for use with WC2&EAST.EXE.

For conversion between various single-byte (8-bit) coding standards, use the parallel utility WC2&8BIT.EXE in the CONVERT subdirectory.

See Also: Eight-bit Code Conversions

Twenty-four-bit Code Conversions

UTF-7 and UTF-8 Conversions

Other DBCS (double-byte character set) utilities are explained below.

Conversion of fixed-width double-byte (16-bit) files to and from WinCALIS Author

Conversion of Duke Chinese Typist files

Use DCT2UNI.EXE with the table DCT2UNI.TAB to convert Duke Chinese Typist 2.0 coded files to Unicode.

In order to access this utility, do the following:

- Go to a DOS prompt and change to "CONVERT" subdirectory
- At a DOS prompt type the command line:
dct2uni sourcefile targetfile [/n] [/r] [/t:tablefile]

The /n switch ignores all file headers (such as the WinCALIS/Uniedit Unicode file header), in both reading and writing files.

The direction of conversion is from a DCT source file to a WinCALIS Author Unicode target file by default. The /r switch reverses the direction of the conversion, that is, from a WinCALIS Author source file to a DCT target file.

The /t:tablefile switch causes the specified table file to be used for conversion. If no table name is specified on the command line, DCT2UNI.EXE will use any table it finds in the current directory with the name DCT2UNI.TAB. If a table name is specified, DCT2UNI.EXE will use that table instead.

There is limited support for DOS wild cards and paths, e.g.,

```
DCT2UNI *.DC2 NEW
```

where 'new' is a subdirectory under the current directory to which converted versions of all files with matching filename *.DC2 will be written. Or:

```
DCT2UNI *.DC2 *.UNI
```

where all source files with file extension .DC2 are converted to target files with file extension .UNI.

Conversion of Miscellaneous Fixed-width 16-bit files

Types of fixed-width 16-bit codes you may wish to convert include the following:

- conversion of text files from various other East Asian word-processing programs
- conversion of older or non-standard Unicode-format files, e.g., of Unicode 1.0 or 1.1-coded files to Unicode 2.0

Use the table U10&U11.TAB and/or the table U11&U20.TAB, or modified versions of them (see below).

- conversion between simple-style Chinese and complex-style Chinese.
Use the tables SMP2CMP.TAB or CMP2SMP.TAB.

Note: Since a simple-style-->complex-style Chinese conversion is one-->many and no intelligence is employed, it will require post-editing by someone familiar with proper traditional usage.

- conversion of files with Korean KS Hanja Compatibility Codes to "pure Unicode"
Use the table KORCOMP.TAB.
- conversion of files in the WinCALIS/UniEdit full code set (including private codes, "compatibility" character codes, and Arabic-script Presentation Characters) to "pure Unicode"
Use the table WC&PURUN.TAB.

Note: WinCALIS maintains some distinctions among Han characters which are not maintained in the Unicode Han character set, by mapping one of the glyphic forms to a code in part of the private-use code space, in the range F000-F1FF. It is our desire to maintain distinctions of significance to language teachers and students within Unicode plain text (without formatting information) and without "escape codes" in the character stream, indicating that the following character is to be understood as "Japanese," or "Korean," etc. This affects approximately 400 Han characters, usually glyphic variations between the Chinese and Japanese character sets or the "standard" P.R.C. and Taiwan character sets, such as the distinct simple-style Chinese and Japanese *jōyōkanji* glyphs for U+4E0E, virtually all characters with the "silk radical" (Kangxi radical 120), and most characters with the "bone" radical (Kangxi radical 188). Here is a sample of the code equivalences (see the file "wc&purun.lst" in the ..\convert subdirectory for the complete list):

WinCALIS	"Pure"
<u>Unicode</u>	<u>Unicode</u>
F001	4E0E
F002	50B2
F003	533E
F004	5448
F005	548E
F006	55F7
...	

When you convert files to "pure" Unicode, these distinctions are lost and cannot be recovered.

In addition, certain other composite characters are also mapped to private-use codes, such as Cyrillic vowels with acute accents (F700-F715), Roman letters with exotic diacritic combinations (F600-F6FF), etc. These characters cannot yet be decomposed automatically during conversion into their component parts.

Single files can be converted by opening or saving them in WinCALIS Author and selecting "Other conversions" in the "List files of type" or "Save file as type" box. Or they may be converted by copying them to the Windows clipboard from the source application, then pasting them into WinCALIS Author using the Paste As... option in the Edit menu to convert codes "on the fly" during pasting, and saving them as WinCALIS Unicode files.

To convert many files at once, you may use the DOS utility WC2&16BT.EXE at the DOS prompt.

WC2&16BT.EXE can be found in the CONVERT subdirectory under your WinCALIS program directory. In order to access this utility, do the following:

- Go to a DOS prompt and change to "CONVERT" subdirectory
- At a DOS prompt type the command line:
wc2&16bt sourcefile targetfile [/n] [/r] [/t:tablefile]

The /n switch ignores all file headers (such as the WinCALIS/UniEdit Unicode file header), in both reading and writing files.

The direction of conversion is from a non-Unicode 2.0 16-bit source file to a WinCALIS Unicode target file by default. The /r switch reverses the direction of the conversion, that is, from a WinCALIS source file to a non-Unicode 2.0 16-bit target file.

The /t:tablefile switch causes the specified table file to be used for conversion. If no table name is specified on the command line, WC2&16BT.EXE will use any table it finds in the current directory with the name WC2&16BT.TAB. If a table name is specified, WC2&16BT.EXE will use that table instead.

There is limited support for DOS wild cards and paths ,e.g.,

WC2&16BT *.U10 NEW

where 'new' is a subdirectory under the current directory to which converted versions of all files with matching filename *.GB will be written. Or:

WC2&16BT *.U11 *.U20

where all source files with file extension .U11 are converted to target files with file extension .U20.

You can modify any of the binary *.TAB tables above or construct a new one with the included utility LST2TABH.EXE by modifying or imitating the *.LST text source files of the above *.TAB files. Each *.LST file has one code correspondence per line, with the codes on each line separated by tabs, e.g.,

```
4E26(tab)5E76
.
.
EAA4(tab)7199.
```

Usage: LST2TABH LISTFILE.LST

See Also: Eight-bit Code Conversions
Twenty-four-bit Code Conversions
UTF-7 and UTF-8 Conversions

Twenty-four-bit Code Conversions

Twenty-four-bit (triple-byte) fixed-width coding standards include CCCII and EACC/ANSI Z39.64. You may convert files using these codes to WinCALIS files with 16-bit Unicode UCS-2 character codes in one of the following ways.

Single files can be converted by opening or saving them in WinCALIS Author and selecting “Other conversions” in the “List files of type” or “Save file as type” box. Or they may be converted by copying them to the Windows clipboard from the source application, then pasting them into WinCALIS Author using the Paste As... option in the Edit menu to convert codes “on the fly” during pasting, and saving them as WinCALIS Unicode files.

To convert many files in “batch mode” rapidly at once, you may use the DOS utility WC2&24BT.EXE at the DOS prompt.

WC2&24BT.EXE can be found in the CONVERT subdirectory under your WinCALIS program directory. In order to access this utility, do the following:

- Go to a DOS prompt and change to “CONVERT” subdirectory
- At a DOS prompt type the command line:
wc2&24bt sourcefile targetfile [/n] [/r] [/t:tablefile.tab]

The /n switch ignores all file headers (such as the WinCALIS/UniEdit Unicode file header), in both reading and writing files.

The direction of conversion is from a 24-bit (3-byte) coded source file to a WinCALIS Unicode target file by default. The /r switch reverses the direction of the conversion, that is, from a WinCALIS source file to a 24-bit target file.

The /t:tablefile switch causes the specified table file to be used for conversion. If no table name is specified on the command line, WC2&24BT.EXE will use any table it finds in the current directory with the name WC2&24BT.TAB. If a table name is specified, WC2&24BT.EXE will use that table instead.

There is limited support for DOS wild cards and paths ,e.g.,

```
WC2&24BT *.EAC NEW
```

where 'new' is a subdirectory under the current directory to which converted versions of all files with matching filename *.EAC will be written. Or:

WC2&24BT *.EAC *.UNI

where all source files with file extension .EAC are converted to target files with file extension .UNI.

Key to supplied conversion tables to be used with WC2&24BT.EXE, (rename as WC2&24BT.TAB or type filename after /t:tablefile switch):

CCCI&UNI.TAB CCCII codes (Taiwan Chinese)
EACC&UNI.TAB EACC/ANSI Z39.64 (REACC) codes (library MARC records, such as Research Libraries Network, Library of Congress etc.)

You can modify any of the binary *.TAB tables above or construct a new one with the included utility LST2TAB6.EXE by modifying or imitating the *.LST text source files of the above *.TAB files. Each *.LST file has one code correspondence per line, with the codes on each line separated by tabs, e.g.,

```
213021(tab)4E00
.
.
70775D(tab)9B0F.
```

Usage: LST2TAB6 LISTFILE.LST

See Also: Eight-bit Code Conversions
Sixteen-bit Code Conversions
UTF-7 and UTF-8 Conversions

UTF-7 and UTF-8 Conversions

You may convert Unicode files with codes in the “Universal Coded Character Set Transformation Format” UTF-7 (7-bit codes) or UTF-8 (8-bit codes) to WinCALIS files with 16-bit UCS-2 character codes in one of the following ways.

Single files can be converted by opening or saving them in WinCALIS Author and selecting “UTF-7/UTF-8” in the “List Files of Type” or “Save File as Type” box.

To convert many files in “batch mode” rapidly at once, you may use the DOS utility UCS2&UTF.EXE at the DOS prompt.

UCS2&UTF.EXE can be found in the CONVERT subdirectory under your WinCALIS program directory. In order to access this utility, do the following:

- Go to a DOS prompt and change to “CONVERT” subdirectory
- At a DOS prompt type the command line:
ucs2&utf sourcefile targetfile [/n] [/r] [/7]

The /n switch ignores all file headers (such as the WinCALIS/UniEdit Unicode file header), in both reading and writing files.

The direction of conversion is from a Unicode UTF-7 or UTF-8 coded source file to a WinCALIS Unicode UCS-2 coded target file by default. The /r switch reverses the direction of the conversion, that is, from a WinCALIS source file to a UTF-7 or UTF-8 target file.

The /7 switch, used in conjunction with the /r switch, converts from UCS-2 to UTF-7. By default conversion will be to UTF-8.

There is limited support for DOS wild cards and paths ,e.g.,

UCS2&UTF *.UTF NEW

where 'new' is a subdirectory under the current directory to which converted versions of all files with matching filename *.UTF will be written. Or:

UCS2&UTF *.UTF *.UCS

where all source files with file extension .UTF are converted to target files with file extension .UCS.

See Also: Eight-bit Code Conversions

Sixteen-bit Code Conversions

Twenty-four-bit Code Conversions

Conversion of Commands in Older WinCALIS Scripts

Aside from character code conversions, there are other typical conversion problems with WinCALIS 1.x scripts. Here are some of them:

- <CUR:r,c> commands need to be reviewed with the awareness that there are 1-2 fewer rows on the WinCALIS 2.x screen (ca. 19 total "Medium" lines).
- <SIZE t,l,b,r> commands are expressed as screen percentages since WinCALIS 1.1, as opposed to Text CALIS and WinCALIS 1.0 row,column numbers. For quick approximate conversions of row and column numbers in old scripts to percentages (based on a 24 row tall, 80 column wide screen), multiply the row number by 4 and the column by 5/4. E.g., a Text CALIS command like <SIZE F4:4,51,20,75> is roughly equivalent to <SIZE F4:16,64,80,94>.
- <TAB:n> commands need to be checked and revised where needed to a ratio of 8:5 (multiply n by 1.6 and round to nearest integer); <TAB:1> is usually OK as is.
- Wild card characters used literally in anticipated responses, preceded by a backslash (e.g., *) usually must be preceded by two backslashes in WinCALIS 2.x (e.g., *).
- Commands expressed as pixel coordinates, like <VIDEO LOC x,y> or <VIDEO SIZE AREA t,l,b,r> need to be converted to screen percentages--divide the first (and third) number by 4.8 and divide the second (and fourth) number by 6.4 (based on standard VGA dimensions of 480 pixels vertical by 640 pixels horizontal).
- Audio and video subsidiary files (*.TND, *.AUD, and *.VID) need to be renamed as *.MMD files, and their internal syntax needs to be converted to WinCALIS 2.x standard:

[deviceorfilename1]

clipname1=begin-time-or-frame, end-time-or-frame

[deviceorfilename2]

clipname1=begin-time-or-frame, end-time-or-frame etc.

The Windows Macro Recorder file CONV.TND.REC in the CONVERT subdirectory can be used to automate this conversion: Run "Recorder" in the Windows "Accessories" group.

- Video commands must be revised. For example, a WinCALIS 1.x string of video commands such as:
<VIDEO: CUE entire1> <VIDEO: PLAY entire2>
should be shortened to just:
<VIDEO: PLAY entire>
- L statements can be deleted, or converted into default font L statements (e.g., LArial).

Appendix A: CALIScript Language Quick Reference

STATEMENTS

Every line in a WinCALIS script must begin with one of the following statement marker characters:

@	Default	As needed - may occur anywhere in script
#	Task label	One per task - must occur in every task (max. 15 characters)
K	Keyboard	One per script - may occur anywhere, but will take effect at start
L	Default font	One per script, e.g., LArial - may occur anywhere, but will take effect at start
C	Comment (Lesson Description)	One per line of lesson description
!	Directions	One per line of directions - may occur in any task
/	Text	One per line of text - may occur in any task
?	Question	One per line of question - must occur in every task but #0 or #end
%	Variable	As needed - variable declaration and initialization, of form %stringvariable1\$ = "value" or %numvariable = integer value
+	Correct/Continue answer	One per Correct/Continue answer - at least one must occur in every task
-	Incorrect/Remain Here answer	One per anticipated error - may occur in any task
=	Neutral rep. response	One per anticipated neutral response - may occur in any task (re-execute every statement in task afterwards)
\$	Neutral no-repeat resp.	One per anticipated neutral response - may occur in any task (do not re-execute statements in task afterwards)
&	Spillover	As needed - may follow +, -, =, or \$ statements
	#0	This is the first task executed in every WinCALIS script.
	#end	This is the last task executed in every WinCALIS script.

WILD CARDS

Within Correct/Continue answer and anticipated Incorrect/Remain Here statements, the following characters have special significance.

^	Matches any single character
[]	Matches any one of the enclosed characters
[n1-n9]	Matches any one of the enclosed characters in Unicode sequence n1 .. n9
*	Matches any sequence of characters within word
&	Matches any sequence of characters, including blanks and null character
 	Matches alternatives separated by vertical bar
{ }	Matches any one of enclosed possibilities (separated by vertical bars)

OTHER RESPONSE LINE SYMBOLS

;	Separates anticipated student response from corresponding feedback. Used in Correct/Continue (+) Incorrect/Remain Here (-) and neutral (=, \$) response statements.
\	Allows literal use of ; and wild card characters within student response. Place \ before the character. Example: The statement +\&;(feedback) will reply 'Correct' and give appropriate feedback if the student enters the character & (and only &).

SPECIAL RESPONSES

Special responses occur in the student answer portion of anticipated response statements. All CALIS special responses, commands, and options are composed of one or more keywords in capital letters, which are surrounded by opening and closing angle brackets (< >). Keywords must be separated by one or more delimiters. Delimiters include the space character (), the colon (:), the semicolon (;), and the comma (,). Multiple delimiters are permitted.

- <BUTTON name [CLICKED|CHECKED|UNCHECKED]>** Defines action by WinCALIS if student clicks (checks, doesn't check) hot button created with <BUTTON> command.
- <BUTTON name1 DRAGDROP name2>** Defines action by WinCALIS if student drags and drops hot button name1 created with <BUTTON> command onto hot button name2.
- <ENTER>** Accepts pressing the [Enter] key (with no other characters typed) as a student response. May be used in a Correct/Continue answer (+) or anticipated Incorrect/Remain Here response (-), and may be followed by a separator (;) and appropriate feedback.
- <GIVEUP>** Defines action by WinCALIS if student ends task with the Giveup key ([F6]). Must be used in an anticipated Incorrect/Remain Here (-) statement, and should be followed by a separator (;) and appropriate feedback.
- <HOTSPOT name [0,0,100,100]>** Defines action by WinCALIS if student clicks on hot area of a graphic. By default 100%, restricted to a smaller area by parameters.
- <IF:response\$=variable\$>** Defines action by WinCALIS if the student's text response, stored in response\$, matches the value of variable\$,
- <IF:mark\$=variable\$>** or if text marked, stored in mark\$, matches the value of variable\$,
- <IF:numvariable LT|=|GT value>** stored in mark\$, matches the value of variable\$, or expression numvariable=value returns true. Valid operators are 'LT' (less than), '=' (equal to), and 'GT' (greater than).
- <KEY 'x'>** Defines action taken immediately by WinCALIS if student presses the 'x' key, without pressing [Enter].
- <KEY SHIFT 'x'>** or presses the shifted 'x' key,
- <KEY CONTROL 'x'>** or presses the 'x' key with the Ctrl key depressed,
- <KEY CONTROL SHIFT 'x'>** or presses the 'x' key with the Ctrl+Shift keys depressed. 'CONTROL' and 'SHIFT' not case-sensitive.
- <KEY ANY>** Defines action taken immediately by WinCALIS if any key is pressed.
- <KEY controlkey>** Defines action taken immediately by WinCALIS if one of the following control keys is pressed: SHIFT, CONTROL, ALT, BACK, TAB, CAPITAL, ESCAPE, SPACE, PRIOR, NEXT, END, HOME, LEFT, UP, RIGHT, DOWN, DELETE, NUMPAD0, NUMPAD1, NUMPAD2, NUMPAD3, NUMPAD4, NUMPAD5, NUMPAD6, NUMPAD7, NUMPAD8, NUMPAD9, MULTIPLY, ADD, SEPARATOR, SUBTRACT, DECIMAL, DIVIDE, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, NUMLOCK
- <MOUSE mark1 mark2>** Defines action by WinCALIS if student clicks on the hot rectangular area with upper left corner at mark1 and lower right corner at mark2.
- <OR>** Defines action by WinCALIS if any special response in a series separated by <OR> operators is true. E.g., +<MOUSE yes1

yes2><OR><KEY 'y'>;feedback. May not be used in combination with text responses.

<PASS>

Defines action by WinCALIS if student passes a task with one of the Pass keys (gray [+], Ctrl+PageDown, gray [-], Ctrl+PageUp). Must be used in an anticipated Incorrect/Remain Here (-) statement, and should be followed by a separator (;) and appropriate feedback.

<PASSBACK>

Defines action by WinCALIS if student passes back on a task with Pass Back key gray [-] or Ctrl+PageUp or the Pass Back menu item). Must be used in an anticipated Incorrect/Remain Here (-) statement, and should be followed by a separator (;) and appropriate feedback.

<+KEY>

Accepts pressing a Pass key, gray "+" on the numeric keypad or Ctrl+PageDown, as a student response. May be used in Correct/Continue answer (+) or anticipated Incorrect/Remain Here answer (-) statements and may be followed by a separator (;) and appropriate feedback. May not be used in tasks which contain <PASS> or <PASSBACK> special responses.

<-KEY>

Accepts pressing a Passback key, gray "-" on the numeric keypad or Ctrl+PageUp, as a student response. Same usage as <+KEY>

COMMANDS

The following commands may occur in directions (!), text (/) and question (?) statements, and in the response portion of Correct/Continue answer (+), Incorrect/Remain Here answer (-), and neutral response (=, \$) statements. See Special Responses above for command syntax.

In the following, [W] indicates an optional window designation keyword which may or may not occur in the command. Accepted window keywords here include QUESTION, TEXT, D, DIR, or F1 .. F10. (Do not include the brackets [] in actual CALIS commands.) Abbreviations are marked with curly brackets ({}).

<ADD: one, two, ..>	Add tasks one, two, .. to secondary task list
<ADD: 1--5.6>	Add tasks 1 through 5.6 (physical order) to secondary task list
<ASSIGN: a=b>	Assign the value b to the variable a
<ASSIGN: n=n+1>	Increment variable n
<AUDIO: XXX>	Audio command {<AUD>}
<AUDIO: clipname>	Play audio clip clipname
<AUDIO: DELETE file.wav>	Delete audio file
<AUDIO: LEFT>	Play left channel
<AUDIO: PAUSE>	Pause audio
<AUDIO: PLAY filename.wav>	Play digital audio wave file
<AUDIO: RECORD file.wav millisecs>	Record student
<AUDIO: RESUME>	Resume audio
<AUDIO: RIGHT>	Play right channel
<AUDIO: STEREO>	Play both channels (stereo)
<AUDIO: STOP>	Stop audio
<BEEP [# of beeps]>	Make a beeping noise from 2 to 10 times
<BLACK>	Black color {<BLK>}
<BLANK [W]>	Erase current window (or window W) {<BLA>}
<BLUE>	Blue color {}
<BOLD>	Toggle on and off bold screen attribute {<BO>}
<BROWN>	Brown color
<BUTTON name CREATE,"Label" [NORMAL w,h]>	Create button 'name' with ANSI 'Label' (max. 55 characters)
	[w pixels wide, h pixels high] 'Name' max. 9 characters.

<BUTTON name CREATE,"Label" STATIC OWNERDRAW w,h>	Create button 'name' with Unicode "Label" (max. 55 chars.) w pixels wide, h pixels high]
<BUTTON name CREATE,"Text" CHECKBOX GROUPBOX RADIOBUTTON w,h>	Create check box, group box, or radio button 'name' with ANSI "Text" w pixels wide, h pixels high]
<BUTTON name FCOLOR BCOLOR COLORNAME>	Assign color to button
<BUTTON name FLASH n>	Flash button n times
<BUTTON name STATE ENABLE DISABLE SHOW HIDE CHECK>	Set state of hot button
<CENTER>	Center the current line of text on-screen {<CE>}
<CLEAR [W]>	Erase current window (or window W) from cursor onwards {<CLE>}
<CLEAR [W] LINE>	Erase line from cursor position to right margin
<CLEAR [W] LINE LEFT>	Erase line from cursor position back to left margin
<CLOSE: window>	Close DIR or tutorial window Fn opened with <POPUP [W] NOLOCK> command
<COMMENT>	Ignore rest of current line in generating *.WCL file
<CORRECT[length][#task]>	Puts the first Correct/Continue Anticipated Response on the screen {<COR>}
<CURSOR [W]: #r, #c>	Set cursor in current window (or window W) to specified row number #r and column number #c{<CUR>}
<CURSOR [W]: +#r, -#c>	Move cursor in current window (or window W) plus or minus #r row units and #c column units from current cursor position {<CUR>}
<CYAN>	Cyan color
<D>	Switch to D window and set it to pop up at bottom of screen
<DELAY: #>	Delay # seconds before continuing
<DELAY: OFF>	Terminate any previously issued DELAY command
<DELETE: one,two,..>	Delete tasks 1,2,.. from task lists {}
<DELETE: 1--5.6>	Delete tasks 1 through 5.6 (physical order) from task lists
<DIR>	Switch to Popup (Directions) Window
<DKBLUE>	Dark blue color
<DKCYAN>	Dark cyan color
<DKGRAY>	Dark gray color
<DKGREEN>	Dark green color
<DKPURPLE>	Dark purple color
<DKRED>	Dark red color
<DONE: tasklabel>	Determine whether task 'tasklabel' has been done and store result in global variable "result"
<DRAW: object_name, object_type[, width, height, border_thickness, border_color, fill_color]>	Draw an object. Parameters: object_name uniquely identifies an object (currently supported object_types are BOX for a box and LINE for a straight line, future object-types 'ARROW', 'DOTTED_LINE', 'DOTTED_BOX'); width: width of the object in pixels (x increment from the current position); height: height of the object in pixels (y increment from the current position); border_thickness: thickness of the border in pixels; border_color: color of the border pixels; fill_color: inside color of box.
<EXEC:program.exe>	Execute program with filename 'program.exe' {<E>}
<EXPORT NEW APPEND filename>	Export contents of current window to a file
<F1> .. <F10>	Switch to indicated tutorial window
<FILL>	Used to generate a blank of specified size or, by default, the size of the longest Correct/Continue answer
<FILL[#][#task][char]>	Print # number of characters specified by char, or print blank based on longest answer in #task
<FONT:>	
<FONTSIZE:size>	Font size change {<FS>}
<FONTSIZE:TINY>	
<FONTSIZE:FINE>	
<FONTSIZE:SMALL>	
<FONTSIZE:MEDIUM>	
<FONTSIZE:LARGE>	
<FONTSIZE:XLARGE>	
<FONTSIZE:HUGE>	
<GLUE filename>	"Glue on" additional WinCALIS task database files to the original WinCALIS script. Often used with <RANDOM>
<GOTO:tasklabel>	Jump to task (used in feedback)
<GRAPHIC:X.bmp [name]>	Display the bitmap graphic file X.bmp [and assign it 'name'] {<GR>}
<GRAY>	Gray color
<GREEN>	Green color {<G>}

<HELPPFILE: myfile.hlp>	Set author-supplied Windows Help file for Author Help (in same directory as script)
<HELPTOPIC: topicid#>	Set topic idnumber in author-supplied Windows Help file
<IF:variable LT = GT value>	Evaluate expression variable=value and execute preceding command or special response if variable is less than, equal to, or greater than value. Valid operators are 'LT' (less than), '=' (equal to), and 'GT' (greater than).
<IMPORT [W]: XX.X>	Import file XX.X to current window or window W {<IMP>}
<INSERT: one, two, ..>	Insert tasks 1, 2, .. into primary task list {<INS>}
<INSERT: 1--5.6>	Insert tasks 1 through 5.6 (physical order) into primary task list
<ITALIC>	Italic font {<IT>}
<LAST>	Print to the screen the last student response
<LASTMARK>	Print to the screen the last text marked by student
<LTGRAY>	Light gray color
<LTGREEN>	Light green color
<LTYELLOW>	Light yellow color
<MARK:xx>	Mark a point in a window to which the cursor can be redirected using the cursor command (CUR:xx) {M:xx}
<MCICTRL: SHOW HIDE>	Show or hide MCI Control Bar--student controls multimedia
<N>	Display sequence number of current task
<NAME>	Insert "Student User" in lieu of name of student
<OLIVE>	Olive color
<ORANGE>	Orange color
<PASS [n]>	Pass forward to next task or forward n tasks
<PASSBACK [n]>	Pass back to previous task or back n tasks
<PEACH>	Peach color
<PINK>	Pink color
<POPHELP>	Pop up author-supplied Windows Help file for Author Help (Help file must be set first with <HELPPFILE: myfile.hlp>)
<POPHELP myfile.hlp [,topicid#]>	Pop up Help file and display topicid#
<POPUP: DIR [NOLOCK]>	Pop up Popup (Directions) Window {<POP>}
<POPUP: Fn [NOLOCK]>	Pop up tutorial window Fn
<PRINT [W]>	Send to default printer contents of current or specified window
<PURPLE>	Purple color {<PUR>}
<REPORT>	Display a detailed task-by-task score report
<QUESTION>	Switch to Interactive (Question) Window {<Q>}
<RANDOM [#] ADD INSERT [r1--rn]>	Randomly ADD or INSERT a task or # tasks, optionally from range r1--rn {<RAND>}
<RANDOM [#] ADD INSERT GLUE [r1--rn]>	Only "glued" tasks will be used {<RAND>}
<RED>	Red color {<R>}
<REVERSE>	Toggle on and off reverse-video screen attribute {<RV>}
<SET parameter [value, OFF]>	Gives certain numeric values that remain in place until changed or removed with another <SET> command.
<SET ANSWER #>	Set number of characters a student may type (used together with <LEN> option)
<SET CORRECT #>	Set number of characters, including trailing blanks, to be printed with <CORRECT> command
<SET FILL #>	Set number of characters, including trailing blanks, to be printed with <FILL> command
<SET Fn [Fn] DISABLE>	Disable Shift+Fn key for listed Fn windows
<SET Fn {Fn} ENABLE>	Enable Shift+Fn key for listed Fn windows
<SET Fn SCROLL ON>	Enable scrolling within listed Fn windows
<SET Fn SCROLL OFF>	Disable scrolling within listed Fn windows
<SET:[W] FONT fontname>	Set font for window to fontname
<SET:LANG langname>	Use "langname.ini" to set pixel heights for standard font sizes for subsequent fonts
<SET LAST #>	Set number of characters, including trailing blanks, to be printed with <LAST> command
<SET:[W] ATTRIBUTE BOLD>	Set window font to bold {<SET:[W] ATTR BO>}
<SET:[W] ATTRIBUTE ITALIC>	Set window font to italics {<SET:[W] ATTR IT>}
<SET:[W] ATTRIBUTE STRIKEOUT>	Set window font to strikeout {<SET:[W] ATTR SO>}
<SET:[W] ATTRIBUTE UNDERLINE>	Set window font to underline {<SET:[W] ATTR UL>}

<SET:[W] ATTRIBUTE [NORMAL]>	Set window font to normal {<SET:[W] ATTR [NO]>}
<SET:[W] BCOLOR color>	Set window background color {<SET: BC color>}
<SET:[W] FCOLOR color>	Set window foreground (text) color {<SET: FC color>}
<SET:[W] FONTSIZE size>	Set window fontsize to size {<SET: FS size>}
<SIZE [W]: a%, b%, x%, y%>	Size current window (or window W) to upper left corner down a%, and over b%, and lower right corner down x%, and over y%
<SIZE [W]: +a%, +b%, +x%, -y%>	Resize current window (or window W) upper left corner down plus or minus a%, over plus or minus b%, and lower right corner down plus or minus x%, and over plus or minus y%
<SIZE TEXT: HIGH>	Automatically size Presentation (Text) Window at top of screen
<SIZE TEXT: LOW>	Automatically size Presentation (Text) Window at bottom of screen
<SPELLMARK>	Mark up spelling of response against first Correct/Continue answer {<SPMK>}
<SPELLCHECK>	(Synonym of <SPELLMARK>) {<SPCK>}
<SPELLMARK:word>	Mark up spelling of response against word or phrase {<SPMK>}
<SUBST:var\$>	Substitute value of variable
<TAB:n>	Tab to nth tab stop
<TEXT>	Switch to Presentation (Text) Window {<T>}
<TOPLINE [W]: #r>	Set row number #r to top of current window (or window W)
<TOPLINE [W]: mark>	Set row of mark to top of current window (or window W)
<TUTNAME>	Display name of tutorial in title bar and in Tutorial menu {<TN>}
<UNDERLINE>	Toggle on and off underline screen attribute {}
<UNGLUE: tasks>	Release glued tasks
<VIDEO>	Video commands {<VID>}
<VIDEO: AUDIO STEREO>	Play both audio tracks
<VIDEO: AUDIO LINE1>	Play track 1
<VIDEO: AUDIO LINE2>	Play track 2
<VIDEO: AUDIO NONE>	Play video without audio
<VIDEO: CUE>	Cue video to starting frame
<VIDEO: FRAME 0,0,100,100>	Crop image to frame percent of image
<VIDEO: HIDE>	Hide video
<VIDEO: LINE1> (<VIDEO:LASER>)	Play video from Source 1
<VIDEO: LINE2> (<VIDEO:VCR>)	Play video from Source 2
<VIDEO: LOC TL>	Show video in top left corner of screen
<VIDEO: LOC TC>	Show video in top center part of screen
<VIDEO: LOC TR>	Show video in top right corner of screen
<VIDEO: LOC CE>	Show video in center of screen
<VIDEO: LOC LL>	Show video in lower left corner of screen
<VIDEO: LOC LC>	Show video in lower center part of screen
<VIDEO: LOC LR>	Show video in lower right corner of screen
<VIDEO: LOC 0,0>	Show video at pixel coordinates 0,0
<VIDEO: LOC mark>	Locate video at mark
<VIDEO: PAUSE>	Pause video
<VIDEO: PLAY clipname>	Play video clip and leave last frame on screen
<VIDEO: PLAYHIDE clipname>	Play video clip, then hide it
<VIDEO: PLAYSND clipname>	Play audio portion of video only
<VIDEO: RESUME>	Resume video
<VIDEO: SHOW>	Show video
<VIDEO: SIZE AREA 0,0,100,100>	Customize area
<VIDEO: SIZE EIGHTH>	Eighth screen
<VIDEO: SIZE FULL>	Full screen
<VIDEO: SIZE QUARTER>	Quarter screen
<VIDEO: SIZE SIXTEENTH>	Sixteenth screen
<VIDEO: STOP>	Stop video
<WHILE: variable LT = GT value>	Execute loop while variable is less than, equal to, or greater than value. Valid operators are 'LT' (less than), '=' (equal to), and 'GT' (greater than).
<WHITE>	White color {<WHI>}
<YELLOW>	Yellow color{<YEL>}

GLOBAL VARIABLES

mark\$ Stores text marked with mouse by student

response\$	Stores text of last student response (Same as <LAST> command)
result	Stores result (0 or 1) of certain commands, such as <DONE>

OPTIONS

These may occur in default (@) or task label (#) statements. Those that occur in default statements govern all tasks that follow. Those that occur in task label statements govern only that task. In the following, the task options marked * are the initial defaults set by WinCALIS. Abbreviations are marked with curly brackets ({}). See Special Responses above for option syntax.

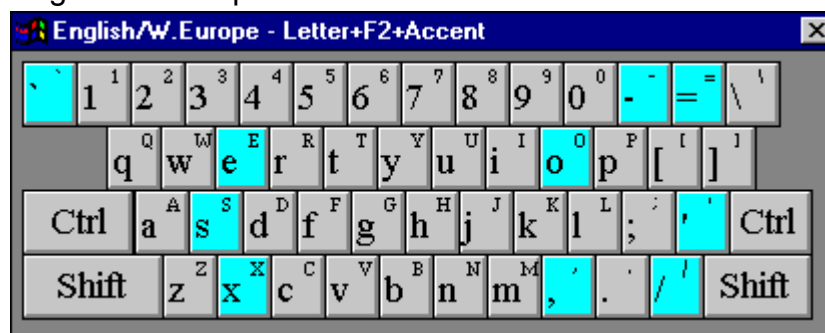
*<ACCENT>	Require correct placement of accents {<ACC>}
<NOACCENT>	Ignore accents and diacritical marks in student responses {<NOA>}
<ALL>	Require student to match all Correct/Continue answers in a task
<ALL:n>	Require student to match n Correct/Continue answers in a task
*<ANY>	Allow student to match any single Correct/Continue answer in a task
<BLANK>	Clear Interactive (Question) Window for each task {<BLA>}
*<NOBLANK>	Leave questions on screen after completion {<NOBL>}
*<CAPITAL>	Require correct capitalization {<CAP>}
<NOCAPITAL>	Ignore capitalization in student responses {<NOCAP>}
*<D>	Show feedback to student responses in the D window
<NOD>	Show feedback to student responses in the Interactive (Question) Window
<GIVE:n>	Allow the student to give up after n tries
*<GIVEUP>	Allow the student to see the Correct/Continue answer to a task by pressing the F6 key if at least one attempt has been made to answer the question.
<NOGIVEUP>	The "give up" feature is disabled. The student cannot be shown the Correct/Continue answer. {<NOGIVE>}
<GROUP:grouptask [,exitgrouptask]>	Identifies group tasks to be executed upon entry to and exit from group
<GROUPLOCK>	Prevents the student from passing out of the current group until all tasks in the group are completed.
*<KEY>	Allow keyboard input
<NOKEY>	Do not allow keyboard input, hide blinking caret
<LENGTH>	Limit number of characters student may type to length of longest Correct/Continue Anticipated Response {<LEN>}
<LENGTH:n>	Limit number of characters student may type to n characters (<LENGTH:0> same as <LENGTH>). {<LEN>}
*<NOLENGTH>	Student may type unlimited characters on screen {<NOLEN>}
*<MAXTRIES:#>	Set a maximum number of student attempts after which the Correct/Continue answer is shown and the task is scored incorrect.
*<MOUSE>	Show mouse hand cursor
<NOMOUSE>	Do not show mouse hand cursor
*<PASS>	Allow students to pass a task {<PAS>}

<PASSBACK>	Same as <PASS> but only goes into effect when student passes backwards with the gray '-' key or Ctrl+PageUp
<NOPASS>	Do not allow students to pass a task {<NOPA>}
*<PRAISE>	Show feedback to student responses (initially, 'Correct' or 'Incorrect') {<PR>}
<NOPRAISE>	Disengage the above feature {<NOPR>}
*<PUNCT>	Require correct placement of punctuation {<PUN>}
<NOPUNCT>	Ignore punctuation in student responses {<NOPU>}
<QUIZ>	Turn on "Quiz" mode, require student to log in and record student responses in disk file 'nameSSno.scr'.
<REPEAT>	Allow completed or deleted task to be reinserted with an <ADD> or <INSERT> command {<REP>}
*<NOREPEAT>	Ignore all further reference to a completed or deleted task {<NOR>}
*<SCORE>	Count a task in final student score {<SCO>}
<NOSCORE>	Do not count a task in final student score {<NOSC>}
*<SHOW>	Show Correct/Continue answer if student gives up
<NOSHOW>	Do not show Correct/Continue answer if student gives up {<NOSH>}

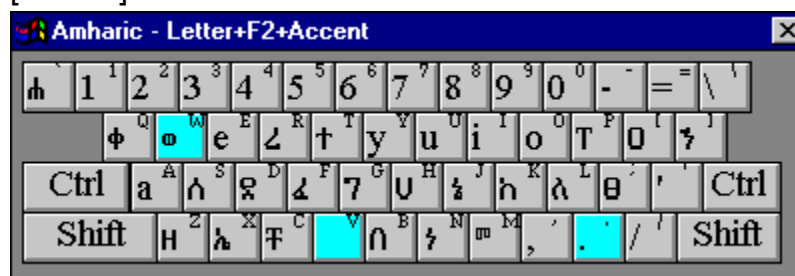
Appendix B: Selected Keyboard Layouts

This section is for referencing the layouts of some selected language keyboards in WinCALIS and WinCALIS Author. Not all the available keyboards are shown. The following keyboards are shown in their unshifted state (unless noted “*shifted*”).

English/W.Europe



[African] Amharic



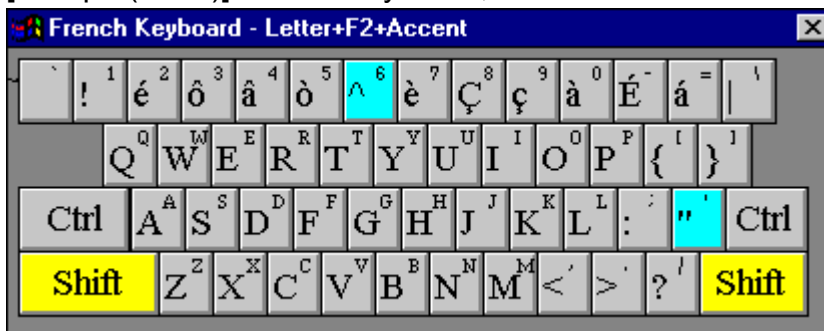
[Arabic script] Arabic-CALIS



[Arabic script] Persian-CALIS



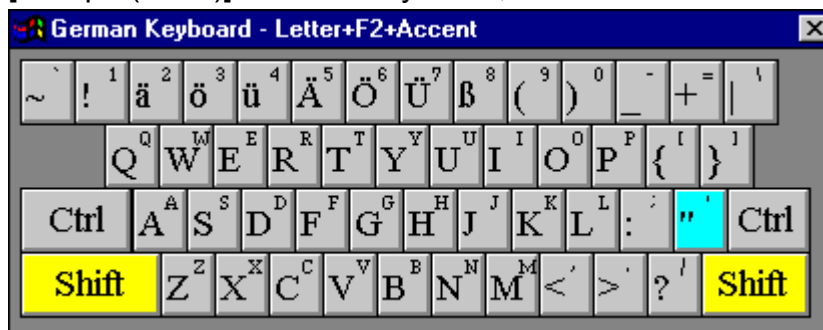
[Europe (other)] Armenian

[Europe (other)] French keyboard, *shifted*

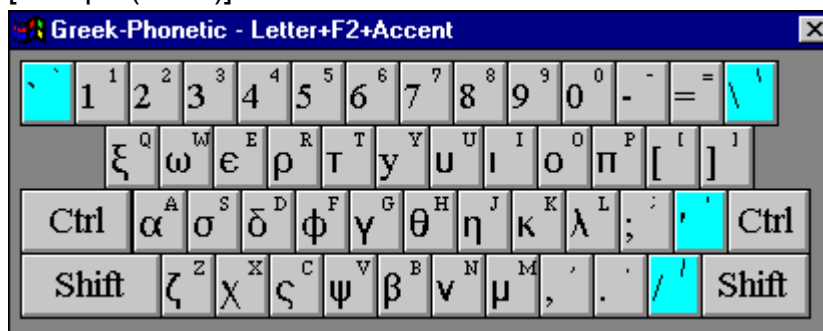
[Europe (other)] Georgian



[Europe (other)] German keyboard, *shifted*



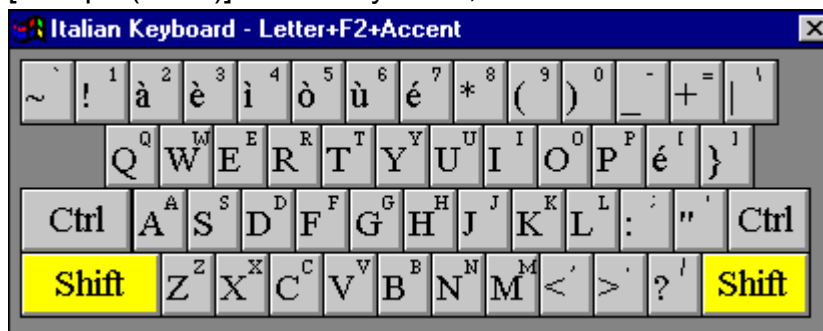
[Europe (other)] Greek-Phonetic



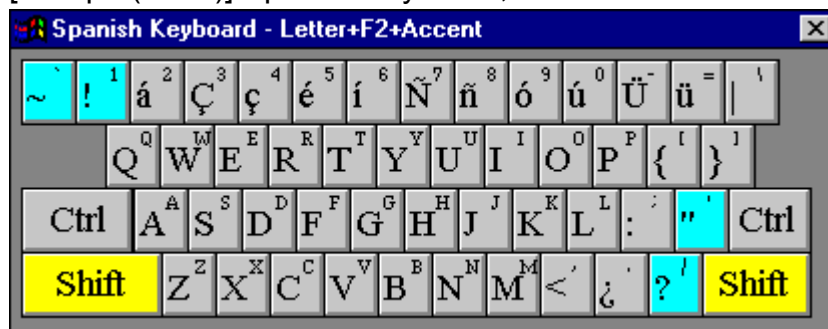
[Europe (other)] Hungarian, *shifted*



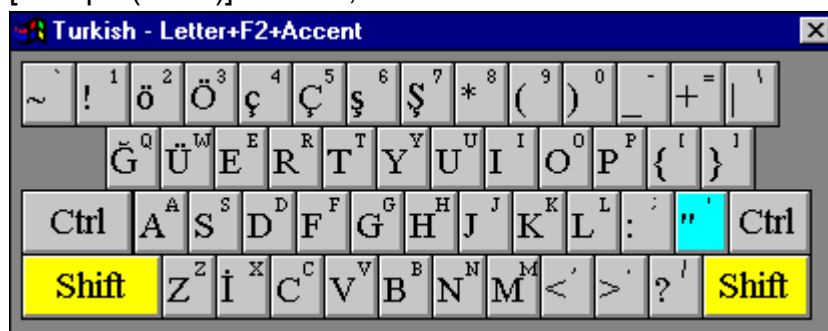
[Europe (other)] Italian keyboard, *shifted*



[Europe (other)] Spanish keyboard, *shifted*



[Europe (other)] Turkish, *shifted*



[Cyrillic] Belarusian



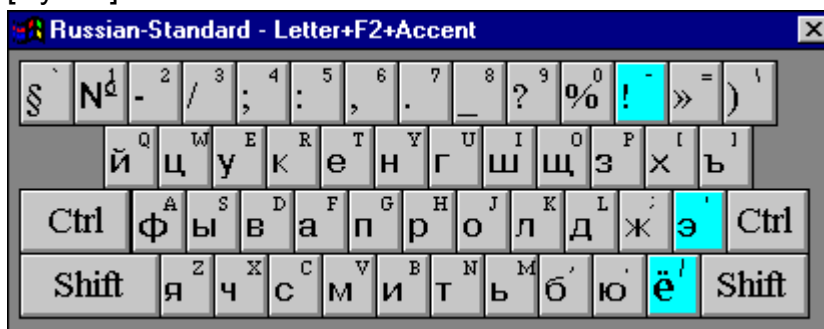
[Cyrillic] Bulgarian



[Cyrillic] Russian-CALIS



[Cyrillic] Russian-Standard



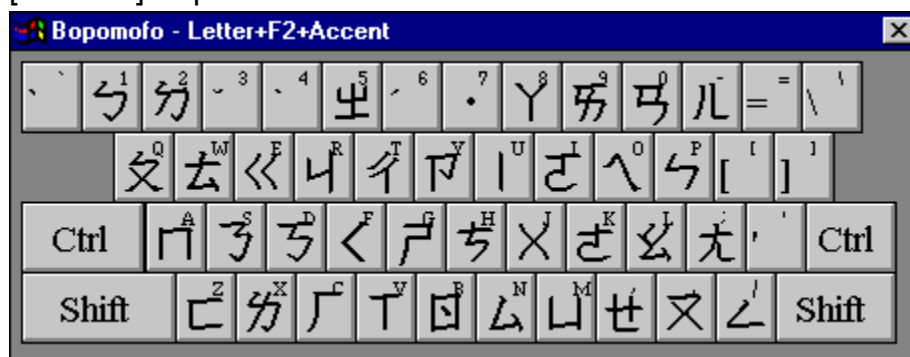
[Cyrillic] Serbian



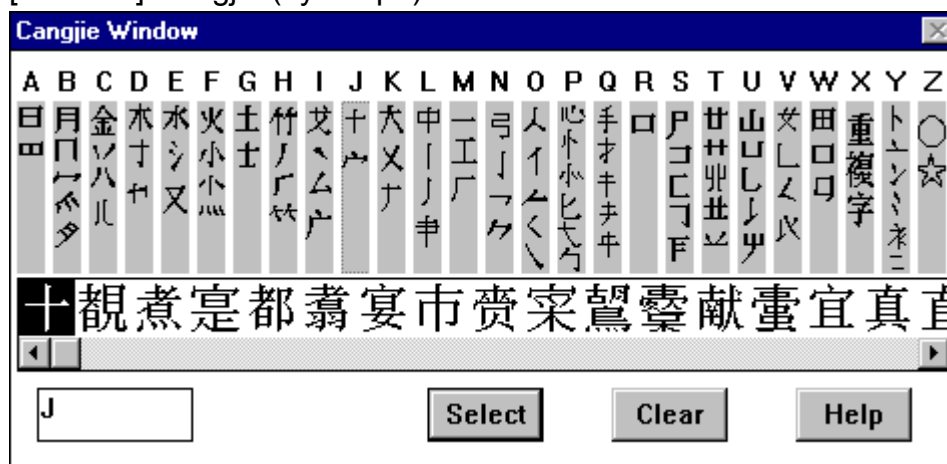
[Cyrillic] Ukrainian



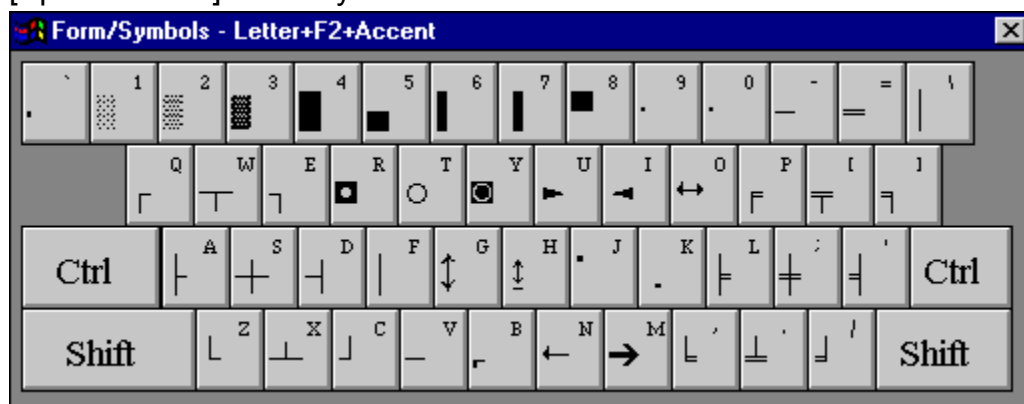
[Chinese] Bopomofo



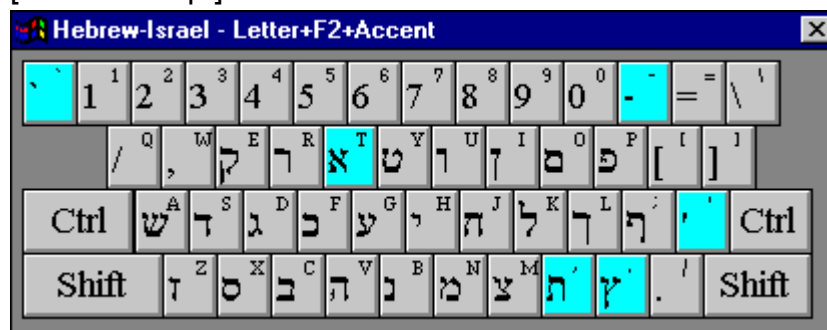
[Chinese] Cangjie (by shape)



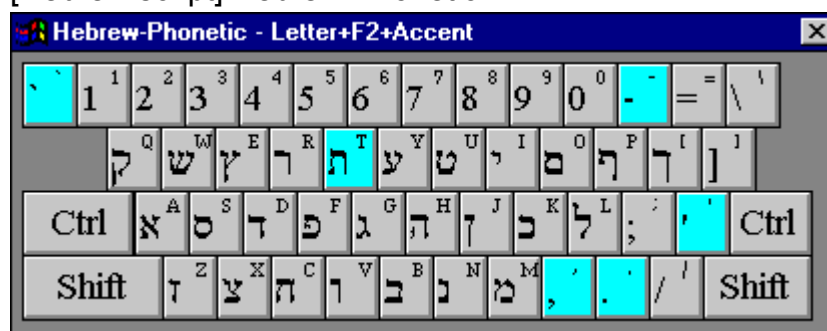
[Special Fonts] Form/Symbols



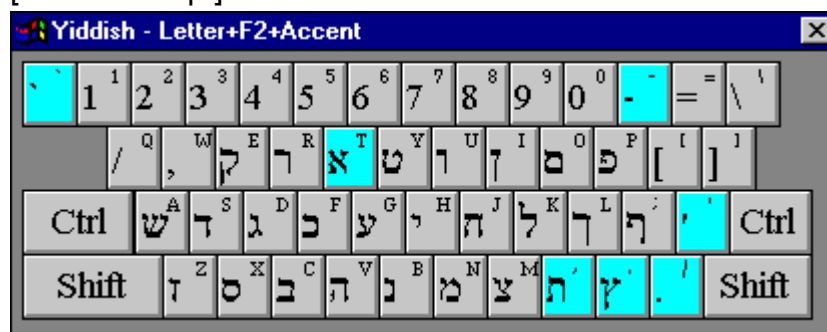
[Hebrew script] Hebrew-Israel



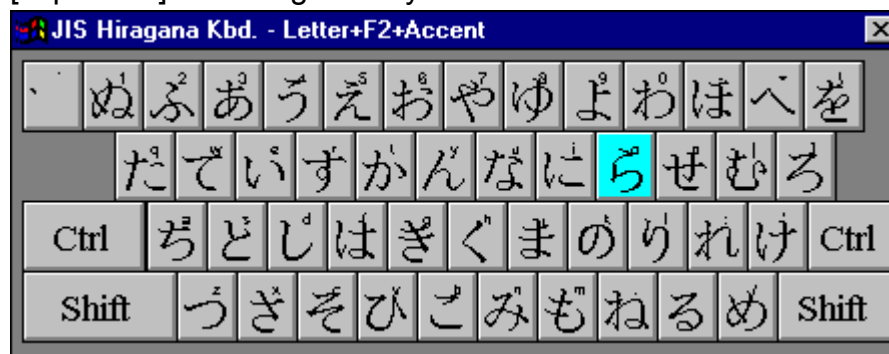
[Hebrew script] Hebrew-Phonetic



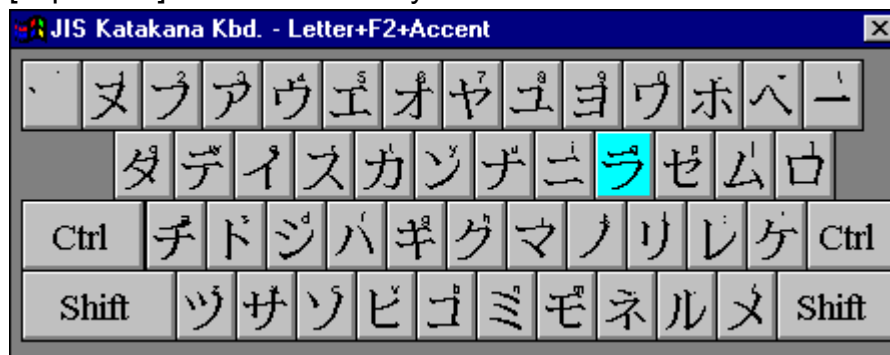
[Hebrew script] Yiddish



[Japanese] JIS Hiragana keyboard



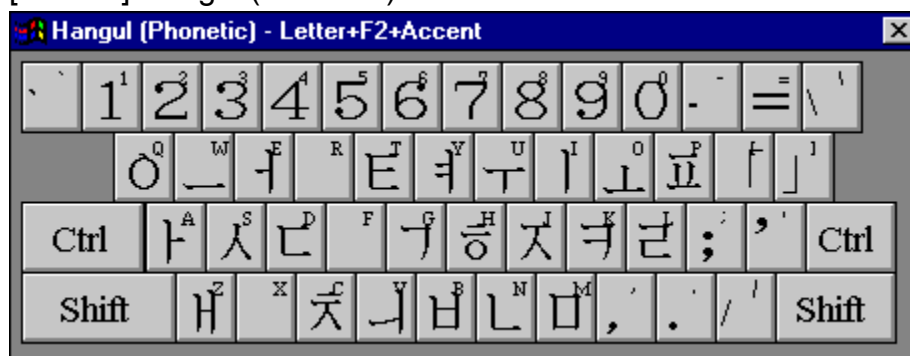
[Japanese] JIS Katakana keyboard



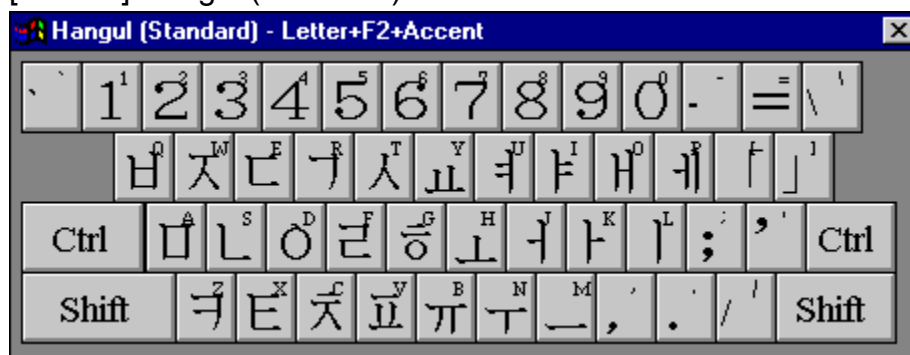
[Korean] Hangul (Kong)



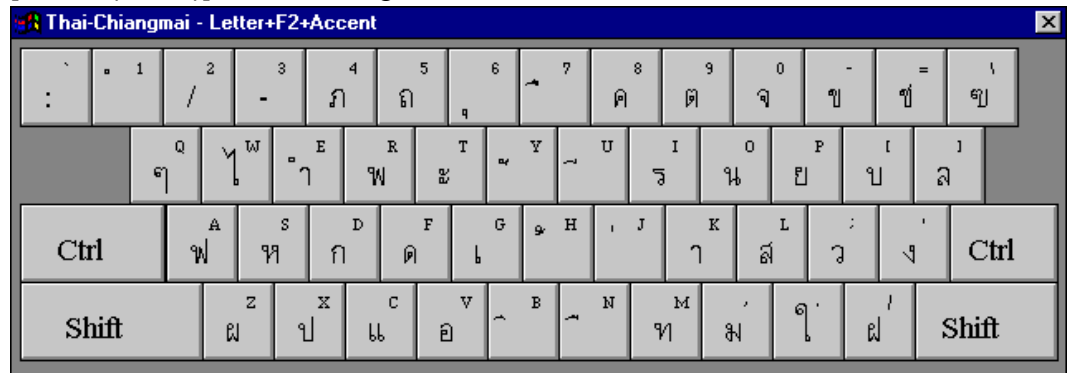
[Korean] Hangul (Phonetic)



[Korean] Hangul (Standard)



[Asian (other)] Thai-Chiangmai



Appendix C: Glossary

“.AVI” file

Digital video file with the file extension *.AVI (standing for “audio-video-interleaved,” which describes how audio and video are stored within the *.AVI file)

“.BMP” file

A Windows standard-format graphics file, with file extension .BMP.

“.CAL” lesson file

Lesson file with the extension “.cal” consisting of straight Unicode text. WinCALIS lessons are produced in this form by the lesson author or by the WinCALIS Author program. Before they can be run, they must be converted to “*.WCL” format.

“.GLU” file

A WinCALIS distribution file format. Task database files which are “glued in” with the CALIS <GLUE> command often end with the extension .GLU. A “.GLU” file is identical in format with a “.WCL” file. In other words, a “.GLU” file is a renamed “.WCL” file which does not appear in the WinCALIS opening screen Lesson Window, and is not intended to be run directly, but rather serve as a task database for other WinCALIS scripts. The Student Workstation version of WinCALIS is only capable of reading *.WCL files, *.WCM files, *.WCI files, *.GLU (glue) files and other such already-converted files.

“.HLP” file

A Windows on-line help file, with the file extension .HLP. A “.HLP” file is opened and viewed with the Windows Help engine, WINHELP.EXE, which can be started by pressing F1 in any Windows application, or with the Program Manager “File,Run” menu option. The WinCALIS Student Workstation includes the following Help files: CALGLOSS.HLP, MLP.HLP, MLPTT.HLP, and WINCALIS.HLP. The Author Workstation also includes CALAUTH.HLP and MLTMEDIA.HLP. In addition, WinCALIS also supports author-created “.HLP” files. See the topic “Author-Supplied Windows Help” in WinCALIS Author on-line Help.

“.INF” file

A WinCALIS table file, with the file extension .INF, usually an editable ASCII file. “WINCALIS.INF,” the source file for WINCALIS.WGI, and “WCFNT.INF” are distributed with the Author Workstation. Some “.INF” files are distributed only in converted binary form as “.WCF” files. See “Customizing and Configuring WinCALIS Author” in on-line Help.

“.LST” file

The editable ASCII version of a WinCALIS conversion filter “.TAB” file. For details on editing “.LST” files, see the topic “WinCALIS Conversions” in WinCALIS Author on-line Help.

“.MAP” file

The editable ASCII text version of a “.WCK” language keyboard file. For details on creating and modifying keyboards, see “Configuring and Customizing WinCALIS Author” in On-line Help.

“.MMD” file

A WinCALIS multimedia clip file. The clips defined in this file are used by the WinCALIS script with the same name: “lesson1.mmd” accompanies “lesson1.cal.” The Author Workstation version of WinCALIS converts the *.MMD ASCII text file into a binary distribution version with the extension .WCM, when it converts the .CAL file into a *.WCL file.

The beginning and ending frame values, or time in milliseconds, of a video or audio clip are stored in a file with the extension “.mmd” in the following manner:

firstclip=340,1899
secondclip=1899,3547

The name of this file is arbitrary with a maximum of eight characters permitted, but it must be the same as the script associated with it. This file is edited automatically by the Multimedia Editor program, though it can also be edited directly. (It is a pure ASCII file.)

“.SUB” file

An optional ASCII text inventory file, which is a list of the various component files of a WinCALIS exercise. Since the files associated with a single exercise can number in the dozens, especially in the case of exercises incorporating graphics and multimedia, some authors make a list of all subsidiary files-- *.BMP graphics files, *.WCI import files, *.GLU glue files, digital audio *.WAV or digital video *.AVI files, etc.--and save it as an ASCII file with the same filename as the exercise and the extension .SUB. They then distribute the “.SUB” file along with the exercise. E.g., the exercise “infinito.wcl” would be accompanied during distribution by the “housekeeping” file “infinito.sub” listing all the subsidiary files required. Since it is annoying for the student or system administrator to receive and try to work an exercise which is missing one or more required peripheral files, good authoring practice encourages keeping track of them by listing all the separate files required for a given exercise in the *.CAL file itself (such as in a header comment in the Pass Through Window), or better, in a separate “.SUB” file (since the *.CAL file is not normally distributed to end-users and the *.WCL file is not directly viewable).

“.TAB” file

A WinCALIS conversion filter file. Generated from an editable ASCII “.LST” file.

“.TUT” file

A common file extension for a WinCALIS Unicode text file, intended to be imported into a WinCALIS Tutorial Window.

WinCALIS automatically converts all tutorial import files into binary distribution “.WCI” files.

“.WAV” file

Microsoft Windows standard waveform digital audio files. They have the file extension .WAV.

WinCALIS only supports playing standard .WAV files in exercises. There are many other standards for digital audio on various platforms and for various sound cards—e.g., Sound Blaster .VOC files, UNIX .AU files (widely used on the Internet), Sounder/Soundtools .SND files, Amiga .IFF files, etc. Files in other standards can usually be converted to .WAV files by means of widely available conversion software, often supplied with sound cards.

“.WCA” author database file

Lesson database file with which the WinCALIS Author program creates a WinCALIS exercise. These files are created and edited using the WinCALIS Author program.

“.WCF” file

A WinCALIS binary table file, with the file extension .WCF. Such files include the six Unicode-to-Font mapping tables, UNI2FNTn.WCF, WCR2L.WCF (Arabic vowel height table), and WCKRN.WCF (Korean Hangul conversion table).

“.WCK” file

A WinCALIS binary language keyboard file. There is one for every language keyboard which appears in the Language menus. It is automatically generated from an editable ASCII “.MAP” file.

“.WCI” file

A WinCALIS distribution file format. When the Author Workstation version of WinCALIS converts a *.CAL script file into a *.WCL file, it also converts any tutorial import files it finds into their binary-encoded equivalent *.WCI files. The “.WCI” file is the distribution form of the tutorial file. The Student Workstation version of WinCALIS is only capable of reading *.WCL files, *.WCM files, *.WCI files, *.GLU (glue) files and other such already-converted files.

“.WCM” file

A WinCALIS multimedia distribution file format. When the Author Workstation version of WinCALIS converts a *.CAL script file into a *.WCL file, it also converts an *.MMD clip file into its binary-encoded equivalent *.WCM file. The “.WCM” file is the distribution form of the clip file. The Student Workstation version of WinCALIS is only capable of reading *.WCL files, *.WCM files, *.WCI files, *.GLU (glue) files and other such already-converted files.

“.WCL” lesson file

WinCALIS lesson file with the extension “.WCL” which is in a binary, secure format, suitable for distribution. The Student Workstation version of WinCALIS is only capable of reading *.WCL files, *.WCM (clip) files, *.WCI (import) files, *.GLU (glue) files and other such already-converted files.

WinCALIS 2.x lessons are written and edited in Unicode text. These lessons have the extension “.CAL.” When WinCALIS runs such a lesson it converts the Unicode text “*.CAL” file to a binary format file with the extension “.WCL.” Once a lesson has been converted, it need not be converted again unless changes are made to the “.CAL” file from which the “.WCL” file was produced.

386 enhanced mode

A mode in which Windows runs to access the virtual-memory capabilities of the 80386 processor. In this mode, Windows appears to use more memory than is physically available and provides multitasking for non-Windows applications. *See also* standard mode, virtual memory.

A

accent key

The key which you press after pressing the F2 overstrike key in order to produce an accented or completely different character. It displays in light blue on the keyboard help file. Click on the key within the keyboard help window or type it from the keyboard and the character displays in red with the accented characters it produces displayed in green.

The window that you are currently using or that is currently selected. Only one window can be active at a time. Keystrokes and commands affect the active window. If a window is active, its title bar changes color to differentiate it from other windows. Windows on the desktop that are not selected are inactive.

ANSI character set

The American National Standards Institute 8-bit character set. It contains 256 characters. The ANSI character set is identical with the first 256 characters of the ISO 10646 Unicode character set.

application

A computer program used for a particular kind of work, such as word processing or database management. This term is used interchangeably with “program.” *See also* Windows application, non-Windows application.

application icon

A graphic that represents a running application. This icon appears only after you start an application and then minimize (reduce) it.

application shortcut key

A key combination that brings a running application to the foreground. You can assign an application shortcut key in Program Manager or in a PIF (program information file).

application swap file

A temporary file that Windows uses to store a non-Windows application when you switch to another application. This leaves more room in memory for running applications. Windows uses application swap files in standard mode only. *See also* swap file.

application window

The window containing the work area and menu bar for an application. The name of the application appears at the top of this window. An application window may contain multiple document windows.

archive attribute

A file characteristic that indicates whether the file has been changed since it was last backed up. This attribute is usually set when the file is created or modified, and then cleared when the file is backed up by using the MS-DOS **xcopy**, **backup**, or **restore** command.

arrow keys

On your computer keyboard, the keys you use to navigate around your screen. Each key is marked with an arrow and is named for the direction in which the arrow points: There IS an UP ARROW, DOWN ARROW, LEFT ARROW, and RIGHT ARROW key. *Also known as* direction keys.

ASCII character set

The American Standard Code for Information Interchange 7-bit character set. It consists of the first 128 (0-127) characters of the ANSI character set (and most other 8-bit character sets) and the first 128 characters of the ISO 10646 Unicode character set. The ASCII character set is the most universal character-coding set.

ASCII text file

A file consisting of only the characters found in the standard ASCII character set and containing no formatting commands or non-displaying characters as found in many word processor-formatted documents.

associate

To identify a filename extension as “belonging” to a certain application. When you open a file with an extension that has been associated with an application, the application starts automatically. Files that are associated with an application are called document files. *See also* document file.

attributes

Information about a file that indicates whether the file is read-only, hidden, or system, and whether it has been changed since it was last backed up. *See also* archive attribute, hidden file, read-only file, system file.

AUTOEXEC.BAT file

A batch program (set of commands) that is automatically carried out by the MS-DOS operating system whenever the computer is started or restarted. The file contains basic startup commands that help configure the system.

automatic link

A link whose Update option is set to automatic. When you edit a linked object, the changes are made to the visual presentation (image) of the object in all documents that contain an automatic link to the object. *See also* manual link, link, object linking and embedding.

B

background

The area behind and outside the active window, including the desktop and any other application windows or icons. *See also* foreground.

background application

An application that is running but is not active. The application cannot receive user input. *See also* foreground application.

batch program

A text file that contains MS-DOS commands. When you run the batch program, MS-DOS carries out the commands in the file as if you had typed them at the MS-DOS prompt.

baud rate

The speed at which information can be transferred through a COM (serial) port.

binary file

A file containing information that is in machine-readable form; it can be read only by an application. In Terminal, these files are not converted or translated during the transfer process.

binary transfer

A method of transferring binary files to a remote computer.

bitmap

An image stored as a pattern of dots (or pixels).

boot

To start or restart your computer, loading the MS-DOS operating system from your hard disk or floppy disk. *See also* hard disk, floppy disk, system disk.

branch

A segment of the File Manager directory tree, representing a directory and any subdirectories within it.

browse

To look through files and directories. Some dialog boxes have a Browse button, which opens another dialog box that lists files and directories so that you can easily select the file you want.

buffer

A temporary storage place for information.

C

CALIS commands

CALIS instructions which display and alter text and screen attributes, cause tasks to be added or deleted, play video and audio from external sources, and basically define the structure of a WinCALIS script. In the CALIS Authoring Language a command is always capitalized and enclosed in angle brackets. *See also* delimiter.

cartridge font

A font contained in a cartridge that physically plugs into your printer. Font cartridges are often used by laser printers such as those in the HP LaserJet family, but some dot-matrix printers also use font cartridges. *See also* font cartridge.

cascade

A way of arranging open windows on the desktop so that they overlap each other, with the title bar of each window remaining visible.

cascading menu

A menu that opens from a command on another menu. A command that opens a cascading menu has a right arrow next to it.

CD Audio

An audio compact disk (CD), played in the CD-ROM drive of a computer. The computer controls the playing of the CD, and the user hears the sound either through headphones connected to the headphone jack of the CD-ROM drive or through the computer's speakers, via a connection between the CD-ROM drive and the computer's sound card.

CD-ROM

A digital mass storage medium, capable of storing approximately 600 megabytes of text, digital audio, or digital video files on a compact disk (CD). Called CD-ROM because they ordinarily can only be read, but not written to.

CD-ROM drive

A peripheral device for playing CD-ROMs.

channel map

Specifies destination channels, output devices, and patch maps for MIDI channel messages.

character set

A group of letters, numbers, and symbols that have some relationship in common. For example, the ASCII character set contains characters that make up the ASCII coding scheme.

check box

A small, square box that appears in a dialog box and that can be selected or cleared. When the check box is selected, an X appears in the box. A check box represents an option that you can turn on or off.

choose

To use a mouse or keyboard to pick an item that begins an action in Windows. You choose commands on menus to perform tasks, and you choose icons to start applications.

clear

To turn off an option by removing the X from a check box. You clear a check box by clicking it, or selecting it and pressing the SPACEBAR.

click

To press and release a mouse button quickly.

client application

An application whose documents can accept linked or embedded objects.

clip

A segment of a longer audio or video recording to be shown to the student in a WinCALIS lesson. The beginning and ending of a clip are marked in milliseconds or frames, which are stored in a clip file, ending with the extension .MMD (ASCII text version) or .WCM (binary distribution version).

Clipboard

A temporary storage location used to transfer data between documents and between applications. Typically, you transfer data to the Clipboard by using an application's Copy or Cut command, and you insert data from the Clipboard by using the application's Paste command. You use Clipboard Viewer to view the contents of the Clipboard.

close

To remove a window or dialog box, or quit an application. You can close a window by using the Close command on the Control menu. When you close an application window, you quit the application.

cloze exercise

A common type of drill in which a student must fill in missing words in a text by guessing the words from context.

clue

In a contextual fill-in-the-blanks cloze exercise, a clue to the correct word or words which fit in the blank. Clues can be either *contextual* (part of the context of the cloze passage) or *noncontextual* (not part of the cloze context).

color scheme

A predefined combination of colors that Windows uses for its screen elements. You can choose a color scheme or create your own by using Control Panel.

command

A word or phrase, usually found in a menu, that you choose in order to carry out an action. You choose a command from a menu, or type a command at the MS-DOS prompt. You can also type a command in the Run dialog box in File Manager or Program Manager. *See also* CALIS commands.

command button

In a dialog box, a button that carries out an action. A command button often has a label that describes the action it carries out (for example, Cancel, Help, or Install). Choosing a command button that is followed by an ellipsis (for example, Browse..) causes another dialog box to appear.

command line

The commands you type to run an application. You can type the commands at the MS-DOS prompt or in the Run dialog box in File Manager or Program Manager.

communications settings

Settings that specify how information is transferred from your computer to a device (usually a printer or modem). *See also* serial port.

compound device

An MCI device that uses media files, compared to a simple device, which does not require media files. For example, to run a compound device such as a MIDI sequencer, you must specify a MIDI file. *See also* Media Control Interface, MIDI.

compound document

A document that contains information created by using more than one application.

CONFIG.SYS file

A text file that contains configuration commands used when you start your computer. Commands in the CONFIG.SYS file enable or disable system features, set limits on resources, and extend the operating-system functionality by loading device drivers.

confirmation message

A message that appears after you specify certain actions, prompting you to confirm that you want to continue with the action or to specify that you want to cancel it. For example, File Manager displays a confirmation message before deleting a file.

control codes

Codes that specify commands or formatting instructions (such as linefeeds or carriage returns) in a text file. Control codes are usually written preceded by a caret (^).

Control menu

A menu that contains commands you can use to manipulate the window. Application icons and some dialog boxes also have a Control menu. To open the Control menu, you use the Control-menu box at the left of the title bar in a window, or you select an application icon. *See also* Control-menu box.

Control-menu box

The icon that opens the Control menu for the window. It is always at the left of the title bar.

Control Panel

The Windows utility program which consists of several mini-applications, or “applets,” allowing you to install fonts, install device drivers, install printers, customize your desktop colors, and so on. It is found in the Program Manager Main group.

conventional memory

Up to the first 640K of memory in your computer. MS-DOS uses this memory to run applications.

copy

To put a copy of the selected text or item onto the Clipboard so that you can transfer it to another location. Most Windows applications have a Copy command on the Edit menu that performs this task. You can also copy entire files from one location to another by using File Manager.

cue

The beginning and end of a video clip—laserdisc or video cassette—to be shown to the student in a WinCALIS lesson. The two halves of a cue share the same arbitrary name with the number 1 or 2 attached to the end to distinguish the beginning or end of the video scene.

current directory

The directory that you are currently working in. In File Manager, the current directory is selected (highlighted) in the directory tree.

cut

To move data from a document into a temporary storage area called the Clipboard. Most Windows applications have a Cut command (on the Edit menu) that performs this task.

cutout

An area of a drawing you select by using the Scissors or Pick tool in Paintbrush.

data file

Any file created within an application: a word-processing document, a spreadsheet, a database file, or a chart, for example. *Also known as* document. *See also* text file.

default button

In some dialog boxes, the command button that Windows automatically selects. The default button has a bold border, indicating that it will be chosen if you press ENTER. You can override a default button by choosing Cancel or another command button.

default printer

The printer that is used if you choose the Print command without first specifying which printer you want to use. You can have only one default printer, which you can specify by using Control Panel or Print Manager. The printer you use most often should be the default printer.

default settings

Settings that are supplied with Windows or WinCALIS. For example, if you print a document from Windows without specifying margin settings, Windows uses its preset margins. Default settings in WinCALIS also refer to the task option settings which are in effect for a WinCALIS lesson assuming that the WinCALIS lesson author has made no changes using the task options. They standardize WinCALIS behavior and allow options such as passing between questions, giving up and seeing the Correct/Continue answer, and requiring proper punctuation and capitalization.

delimiter

In the CALIS authoring language, a symbol which separates keywords in CALIS commands, options, and special responses. CALIS delimiters include the space character (), the colon (:), the semicolon (;), and the comma (,). Keywords must be separated by one or more delimiters. Multiple delimiters are permitted. A common use of delimiters is between a CALIS command, option, or special response keyword and its parameter keywords (or arguments), e.g., the command <SET: F2 DISABLE>, the option <ALL 4>, and the special response <HOTSPOT name 10,10,30,30>. Sometimes delimiters serve to indicate null parameters: in the command <CUR:,2> the comma delimiter indicates that the first “row” parameter is null and “2” is the “column” parameter.

desktop

The screen background for Windows on which windows, icons, and dialog boxes appear.

desktop pattern

A design that appears across the desktop. You can use Control Panel to create your own pattern or choose a pattern provided by Windows.

destination directory

The directory to which you intend to copy or move one or more files.

destination document

A document containing a link to an object in a source document.

device

A component of the system's hardware configuration, such as a modem, printer, mouse, sound card, or disk drive. Also used in the special sense of a media device, which can be either simple or compound

device contention

Occurs when more than one application is trying to use the same device, such as a modem, while Windows is running in 386 enhanced mode.

device driver

Software that controls how a computer communicates with a device, such as a printer or mouse. For example, a printer driver translates information from the computer into information the printer can understand.

device independent bitmap

A standard graphics file format supported by Windows and WinCALIS. "Device independent" means that a graphic can be displayed regardless of the graphics card installed on the computer the student is using. However, graphics created using 256 colors may not look as good on a monitor which only supports 16 colors as graphics originally created for just 16 colors.

Diagnostic Window

A WinCALIS window which appears after a student has entered a response. It contains feedback specified by the WinCALIS lesson author for the particular student answer. Also known as the Directed Feedback Window, or just the "D" Window.

The diagnostic window appears by default under the last line of the question window and is centered on-screen, just large enough to display the text it contains. When the diagnostic window is displayed, WinCALIS pauses until the student presses the space bar or points the mouse at the window itself and clicks.

dialog box

A window which appears temporarily to request information. Many dialog boxes have options you must choose before Windows can carry out a command. In WinCALIS Author Dialog Boxes you can specify text attributes, set lesson defaults, and enter text in both the Roman alphabet and a language you specified using the "L:[Language]" menu option.

dimmed

Unavailable or disabled. A dimmed button or command is displayed in light gray instead of black, and it cannot be chosen. Also known as *grayed out*.

Directions Window

A WinCALIS window which appears by itself ('pops up'). It is usually used to display instructions to the student. When WinCALIS displays the Directions Window the lesson is paused and the message "Press space bar or click mouse here" is displayed until the student either presses the space bar or points the mouse at the window and clicks.

directory

Part of a structure for organizing your files on a disk. A directory can contain files and other directories (called subdirectories). The structure of directories and subdirectories on a disk is called a directory tree. *See also* directory tree, subdirectory.

Directory Description File

A file with the name "WinCALIS.dir" which contains information provided by the WinCALIS lesson author for the student. It is displayed in the Directory Description Window of the opening WinCALIS screen when the student moves into the directory in which contains the file.

Directory Description Window

The window in the lower right-hand corner of the opening WinCALIS screen. It displays the contents of the file "WinCALIS.dir", if such a file exists in the current directory.

directory tree

A graphical display of the directory structure of a disk. The directories on the disk are shown as a branching structure that resembles a tree. The top-level directory is known as the root directory.

Directory Window

The window in the lower left-hand corner of the opening WinCALIS screen containing a listing of all subdirectories below the current directory, followed by a listing of drives on the computer.

disk

A medium for storing information. Information stored on a disk remains there even when you turn your computer off, unlike information stored in memory (RAM). A floppy disk can be inserted and removed from a floppy disk drive, whereas a hard disk is permanently mounted inside its drive. *See also* disk drive.

disk cache

A portion of memory set aside for temporarily holding information read from a disk.

disk drive

A device used for storing and retrieving information on disks. *See also* disk, floppy disk, hard disk.

display adapter

Hardware that converts (adapts) video memory to video output.

document

Whatever you create with an application, including information you type, edit, view, or save. A document may be a business report, a spreadsheet, a picture, or a letter, for example, and is stored as a file on a disk.

document file

A file that is associated with an application. When you open a document file in File Manager, the associated application starts and loads the file.

document window

A window within an application window. There can be more than one document window open at a time.

double-click

To rapidly press and release a mouse button twice without moving the mouse. Double-clicking carries out an action, such as starting an application.

drag

To move an item on the screen by selecting the item and then pressing and holding down the mouse button while moving the mouse. For example, you can move a window to another location on the screen by dragging its title bar.

drive letter

The letter assigned to a drive—for example, C. *See also* disk drive.

dynamic commands

WinCALIS commands which affect the order in which a lesson is executed or cause an external device such as a laserdisc player or Audio CD in a CD-ROM drive to play. Dynamic commands may not appear in Tutorial Windows.

dynamic link library

A Windows file of executable code which is able to be shared by two or more application programs. A dynamic link library has the file extension “.DLL” and is often referred to as simply a “DLL.” WinCALIS and WinCALIS Author make extensive use of shared code in DLLs.

E

embed

To insert information (an object) that was created in one document into another document (most often the two documents were created with different applications). The embedded object can be edited directly from within the document. To embed, you must be using applications that support object linking and embedding. *See also* link, object.

embedded object

Information created in one document and inserted into another document (most often the two documents were created with different applications). Embedded objects can be edited from within the destination document. *See also* linked object.

emulation mode

A mode in which a device such as a printer can emulate the actions of a different device. *See also* terminal emulation.

environment variable

Environment information, such as a drive, path, or filename, associated with a symbolic name that can be used by MS-DOS. You use the MS-DOS set command to define environment variables.

EPT port

A port requiring a special card and software. Only certain printers, such as the IBM Personal Pageprinter, use this port.

exclusive application

A non-Windows application that has the sole use of the computer's resources while it is the active application and Windows is running in 386 enhanced mode. You can specify whether an application gets exclusive use of resources by modifying the application's PIF or by using the Settings command on the application's Control menu. When an exclusive application

is running in a window, it gets most, but not all, of the resources. *See also* PIF.

expand

In File Manager, to show hidden directory levels in the directory tree. You can expand a single directory level, one branch of the directory tree, or all branches at once.

expanded memory

Memory in addition to conventional memory that some non-Windows applications use. Expanded memory is an older standard being replaced by the use of extended memory. Only EMS-compatible software can use expanded memory. Windows running in 386 enhanced mode simulates expanded memory for applications that need it. *Also known as* EMS memory.

expanded-memory emulator

A program that converts extended memory into expanded memory.

extend selection

To select more than one item. For example, in File Manager you can select a group of files to be moved or copied.

extended memory

Memory beyond 1 megabyte (MB) in 80286, 80386, and 80486 computers. Windows uses extended memory to manage and run applications. Extended memory typically is not available to non-Windows applications or MS-DOS. *See also* XMS memory.

Extended Memory Manager

A program that prevents different applications from using the same part of extended memory at the same time.

extension

The period and up to three characters at the end of a filename. An extension usually identifies the kind of information a file contains. For example, files that you create by using WinCALIS Author or the Windows Calendar application have the extension .CAL.

F

file

A collection of information that has been given a name and is stored on a disk. This information can be a document or an application.

file format

The way information is structured in a file. Applications always store data files in a particular format. A format readable by one application may not be readable by another application.

filename

The name of a file. Windows uses MS-DOS naming conventions.

flexible box

In Paintbrush, a square box that changes shape as you move the mouse. You use a flexible box to define an area of the picture you want to change, cut, copy, or move.

floppy disk

A disk that can be inserted in and removed from a floppy disk drive. *See also* hard disk, disk drive.

flow control

In Control Panel and Terminal, the processes used to regulate the rate at which information is transferred from one device to another. One device sends a signal to the other when information can be transferred. *Also known as* handshake.

font

A set of letters, numbers, punctuation marks, and symbols that are a given size and design. *See also* scalable typeface.

font cartridge

A piece of hardware that is plugged into a printer to supply one or more fonts. *See also* cartridge font.

font family

A group designation that describes the general look of a font. For example, the Roman font family contains proportionately spaced fonts with serifs.

footer

Text that appears at the bottom of document pages when they are printed.

foreground

The area of the screen that the active window occupies. *See also* background.

foreground application

The application you are working with. The foreground application appears in the active window. *See also* background application.

format

(1) The way text is set up on a page. (2) To prepare a disk for holding information. Formatting a disk deletes all information that was previously on it. (3) The way information is structured in a file, often specific to one application. *See* file format.

frame value

The value taken from the frame codes on a laserdisc or other video device. It is used to mark the beginning or end of a clip and is stored in the clip file following the clip name with which it is associated.

full-screen application

A non-Windows application that is displayed in the entire screen—rather than a window—when running with Windows. You can specify whether a non-Windows application runs in a full screen or a window by modifying the PIF or using the Control menu. *See also* PIF.

G***General MIDI specification***

Defines a standard setup for MIDI files, including channel settings, instrument patch settings, and key assignments. *See also* MIDI.

F6 - The WinCALIS Give-up key

Unless this function is altered or disabled by the lesson author, the student may press F6 or select “Give Up” from the WinCALIS “Options” pull-down menu to give up and see the Correct/Continue answer, if at least one attempt has been made to answer the question. The question is then scored “incorrect.”

graphics mode

The mode that enables applications to display images in addition to text. Windows applications always run in graphics mode, whereas non-Windows applications can run in either graphics or text mode.

graphics resolution

The level of quality at which Windows prints graphics. The higher the resolution, the better the quality of the printed graphics. Higher-resolution graphics take longer to print and use more disk space.

group

(1) A collection of applications, accessories, or documents within Program Manager. Grouping these items makes them easier to find when you want to start them. (2) In WinCALIS, a group of tasks, which usually share the same Directions. A task group is sometimes preceded by a special Group Task and followed by a Group-Exit Task.

group icon

The icon that represents a group in Program Manager when the group window is minimized (reduced). To open a group and view its contents, you choose the group icon.

group task

A special task inserted in the CALIS script by the Author program to contain directions, whenever directions are written in the Directions Window or the cloze text, when a cloze task is created. The group task is executed automatically whenever WinCALIS enters a group for the first time. Execution of the Directions Window lines as part of the group task ensures that the student always sees the directions intended for a particular group of tasks when he presses F9 or enters the group, regardless of whether he is passing forward or back as the time.

group window

A window that displays the items in a group within Program Manager. These items may be applications, accessories, or documents.

H***hard disk***

A disk that is permanently mounted in its drive. *Also known as* fixed disk. *See also* disk, disk drive, floppy disk.

hardware

The equipment that makes up your computer system, including such items as the keyboard, mouse, disk drives, and monitor.

header

Text that appears at the top of every page of a document when it is printed.

hexadecimal code

A character code expressed as base (radix) 16, rather than base 10, as with decimal codes. Unicode is usually expressed as hexadecimal code. By convention, the digits between 9 and 15 are represented by the letters A-F (or a-f). A hexadecimal code "A" is the same as a decimal code "10"; hexadecimal "F" is decimal "15"; hexadecimal "10" is decimal "16" (i.e., $1 \times 16 + 0 \times 1$); hexadecimal "FF" is decimal "255" (i.e., $15 \times 16 + 15 \times 1$).

hidden file

A file that is not intended to be viewed in a directory listing, such as your Windows swap file. The hidden attribute specifies whether a file is hidden. *See also* attributes.

high memory area (HMA)

The first 64K of extended memory. This area is used by some applications and by Windows.

highlighted

Indicates that an object or text is selected and will be affected by your next action. Highlighted text appears in reverse video on monochrome monitors or in color on some color monitors. Highlighted objects might change color or be surrounded by a selection cursor. You can change the color of the highlight by using Control Panel.

hue

In Control Panel, the position of a color along the color spectrum. For example, green is between yellow and blue. *See also* luminosity, saturation.

I

i/o addresses

Locations within the input/output address space of your computer used by a device, such as a printer or modem. The address is used for communication between software and a device

icons

Graphical representations of various elements in Windows, such as disk drives, applications, embedded and linked objects, and documents.

import

To create a package by inserting an existing file into Object Packager. When you import a file, the icon of the application you used to create the file appears in the Appearance window, and the name of the file appears in the Contents window.

inactive window

Any open window that you are not currently working in. *Also known as* background window.

initialization files (*.INI files)

Files with the extension .INI that contain information that defines your Windows environment. Windows and applications use the information stored in these files.

insertion point

The place where text will be inserted when you type. The insertion point usually appears as a flashing vertical bar in an application's window or in a dialog box. *Also known as* cursor.

interrupt

A signal that a device sends to the computer when the device is ready to accept or send information.

interrupt request lines (IRQ)

Hardware lines over which devices can send interrupts (signals indicating that the device is ready to accept or send information). Typically, each device connected to the computer uses a separate IRQ.

J K

key map

A MIDI patch-map entry that translates key values for certain MIDI messages—for example, the keys used to play the appropriate percussion instrument or a melodic instrument in the appropriate octave.

keyboard buffer

A temporary storage area in memory that keeps track of keys that you typed, even if the computer did not immediately respond to the keys when you typed them.

L

Lesson Description (Comment) Window

The window which appears at the top of the opening WinCALIS screen. It displays a message provided by the author of the lesson that is currently highlighted in the list of WinCALIS lessons. It displays the description of the WinCALIS lesson highlighted in the Lesson Window (if the author has written such a description). Otherwise it displays the message "Welcome to WinCALIS."

Lesson Window

The window which appears in the middle of the opening WinCALIS screen. It displays all of the WinCALIS lesson files in the current directory. If you click on the Options menu and select the menu item "Text Conversion (*.CAL)" (Only in the Author Workstation version of WinCALIS), the Lesson Window will display all of the WinCALIS script files with the extension ".cal" in the current directory.

lexical chain exercise

A common type of exercise in which the student is required to list all of the words in a text which have something in common (i.e., nouns, colors, words with neuter endings).

link (n)

A reference in a destination document to where an object exists in a source document.

link (v)

To create a reference in a destination document to an object in a source document. When you link an object, you are inserting a visual presentation of the object into the destination document. The linked object can be edited directly from within the destination document. When the object changes in the source

document, the changes appear in the destination document. *See also* embed, object.

linked object

In a destination document, a visual presentation (such as an icon or bitmap) of an object.

list box

Within an application window or dialog box, a type of box that lists available choices—for example, a list of all files in a directory. If all the choices do not fit in the list box, there is a scroll bar.

local printer

A printer that is directly connected to one of the ports on your computer. *See also* network printer.

luminosity

In Control Panel, the brightness of a color on a scale from black to white. For example, light blue has a higher luminosity than a darker blue or dark red, but may have about the same luminosity as a light pink. *See also* hue, saturation.

M

macro

In Recorder, a series of recorded actions. You can use Recorder to create macros. When you run a macro, Recorder carries out all the recorded actions.

manual link

A link whose Update option is set to manual. When the linked object is edited, its visual presentation (image) does not reflect the changes until you manually update the link. *See also* automatic link, link, object linking and embedding.

mark

(1) To select text in a non-Windows application so that you can copy it onto the Clipboard. (2) In WinCALIS, selected text to be formatted with a command in the Format menu. (3) In a WinCALIS cloze exercise, text which is highlighted to designate it as a blank for the student to fill in.

Maximize button

The small box containing an up arrow at the right of the title bar. Mouse users can click the Maximize button to enlarge a window to its maximum size. Keyboard users can use the Maximize command on the Control menu.

Media Control Interface (MCI)

A standard control interface for multimedia devices and files. Using MCI, a multimedia application can control a variety of multimedia devices and files. Windows 3.1 originally provided two MCI drivers: one controls the MIDI sequencer, and one controls sound for .WAV files. Other MCI drivers for CD-Audio and digital video are also commonly installed in Windows. The manufacturer of a piece of multimedia-related hardware such as a VCR, a sound card, or a video overlay card must provide an MCI-compatible “driver” for that hardware (or conform to an existing driver), in order for it to be accessible via MCI to all Windows applications. MCI drivers are typically found in the Windows “System” subdirectory, and have filenames of the form “mci*.drv.” There are often other related files in the “System”

subdirectory for a particular device driver.) If a piece of hardware does not support MCI, applications must communicate with it through specially written proprietary drivers. WinCALIS 2.x supports a virtually unlimited range of multimedia hardware and software through the Windows 3.1 Media Control Interface.

media file

A file containing multimedia data, such as sound or animation.

memory

A temporary storage area for information and applications. *See also* RAM, ROM, conventional memory, expanded memory, extended memory.

memory-resident program

A program that is loaded into memory and is available even when another application is active. *Also known as* a terminate-and-stay-resident (TSR) program. *See also* pop-up program.

menu

A list of available commands in an application window. Menu names appear in the menu bar near the top of the window. You open a menu by selecting the menu name. *See also* Control menu.

menu bar

The horizontal bar containing the names of the application’s menus. It appears below the title bar.

MIDI (Musical Instrument Digital Interface)

Acronym for Musical Instrument Digital Interface, a standard protocol for communication between musical instruments and computers. MIDI is the industry standard for electronic music. Sounds are synthesized from brief symbolic information stored in a file on disk, rather than being fully represented digitally, as is the case with digital audio. This results in a very economical use of storage space. MIDI files usually have the file extension *.MID or *.RMI. They are played via the MCI “MIDI Sequencer” driver. WinCALIS exercises can incorporate MIDI audio, either educationally—e.g., playing the national anthem of a country—or as background “mood music,” to accompany the student as he or she works.

MIDI file

A file containing all the information required to play a sound by using a MIDI device.

MIDI sequencer

A program that plays or records sounds stored as MIDI files. Windows provides an MCI MIDI sequencer. *See also* Media Control Interface.

MIDI setup

Specifies the type of MIDI device you are using, its channel mappings, and the port it is connected to.

Minimize button

The small box containing a down arrow at the right of the title bar. Mouse users can click the Minimize button to reduce a window to an icon. Keyboard users can use the Minimize command on the Control menu.

modem

A communications device that enables a computer to transmit information over a telephone line.

monospaced font

A font in which all characters have uniform widths, such as Courier New. *See also* proportional font.

MS-DOS prompt

At the MS-DOS command line, the character or characters that appear at the beginning of the line, indicating that the computer is ready to receive input. *Also known as* command prompt.

multimedia

A combination of various media, such as sound, graphics, animation, and video. Sound includes voice tape recordings played on a cassette recorder connected to an external plug on the computer, or the same voice digitized and saved as audio files stored on the computer's disk. Video includes videotapes and laserdiscs played on a video cassette recorder (VCR) or laserdisc player connected to an external plug on the computer, or the same images digitized and saved as full-motion video files on the computer's disk.

multitasking

A computer's ability to run more than one application at the same time.

N

network

A group of computers connected by cables or other means and using software that enables them to share equipment (such as printers and disk drives) and exchange information.

network drive

A disk drive that is available to multiple users and computers on a network. Network drives often store data files for many people in a work group.

network printer

A printer shared by multiple computers over a network. *See also* local printer.

nonsolid color

A color produced by a pattern of differently colored dots that simulate the desired color. A solid color on one device (such as a 256-color monitor) may be a nonsolid color on another device. *Also known as* dithered color. *See also* solid color.

non-Windows application

An application that is designed to run with MS-DOS but not specifically with Windows and that may not be able to take full advantage of all Windows features (such as memory management).

numeric keypad

The set of numbered keys on the right of the original PC keyboard is called the numeric keypad. In WinCALIS these keys, along with their cursor control key counterparts, are used for various functions including "Repeat previous answer" (= Up Arrow),

"Scroll text" (= PageUp and PageDown), and "Go to beginning/end of line" (= Home/End).

O

object

In the context of object linking and embedding, information such as a drawing, chart, or sound file that can be linked to or embedded into another document. *See also* link, embed.

object linking and embedding (OLE)

A way to transfer and share information between applications. *See also* link, embed, object.

open

To display the contents of a file in a window or to enlarge an icon to a window.

option

(1) A choice in a dialog box. (2) In an MS-DOS command line, a parameter used to control how a command is carried out. Typically, an option starts with a slash (/). For example, to start Microsoft Word in character mode, you type **word /c** at the MS-DOS prompt. The /c is the option. (3) In WinCALIS, the setting which controls how a script or task behaves. *See also* parameter.

option button

A small, round button that appears in a dialog box. Within a group of related option buttons, you can select only one. Also known as *radio button*.

overstrike key (F2)

The key which you press before pressing an accent key in order to produce an accented or completely different character. A message appears in the title bar of the virtual keyboard to remind you: Letter+F2+Accent.

P

package

A small drawing (or icon) that represents an embedded or linked object. When you choose the package, the application used to create the embedded or linked object either plays the object (for example, a sound file) or opens and displays the object (for example, text or graphics).

Pangaea

The legendary continent which preceded all modern continents and united them as one. Pangaea symbolizes WinCALIS's role in recreating the linguistic community which prevailed before Babel, in which all peoples communicated effortlessly. It is also the name of the Local Area Network of the Humanities Computing Laboratory.

parallel interface

An interface between a computer and a printer in which the computer sends multiple bits of information to the printer simultaneously. Parallel and Centronics interfaces are the same type. *See also* serial interface.

parallel port

A connection on a computer, usually LPT1, where you plug in the cable for a parallel printer. Windows supports parallel ports LPT1 through LPT3.

parallel printer

A printer with a parallel interface, which you connect to a parallel port.

parameter

Information added to the command that starts an application. A parameter can be a filename or any type of information up to 62 characters in length. *See also* option.

parity

An error-checking procedure in which the number of 1's must always be the same—either even or odd—for each group of bits transmitted without error.

Pass Through Window

A WinCALIS Author window with special properties. Everything written in the Pass Through Window is passed through directly to the WinCALIS script. It thus provides the experienced author with the means to write statements, commands, or entire tasks, written in the CALIS Authoring Language to the script.

The Pass Through Window is also a good place for the author to write comments and notes during lesson development, since the contents of the Pass Through Window for each task are saved along with the *.WCA author database and included in the *.CAL script file at script generation time. Much like a memo pad for the author, the Pass Through Window will save an author's notes that can be reviewed at a later date; a student will never see the notes contained in a Pass Through Window. Care should be taken that no comment lines begin with capital "C," capital "K," or capital "L," which have special meaning in CALIS syntax. By preceding each comment line with a colon (:) or some other symbol which is not part of the CALIS language, potential confusion can be avoided. For more detailed information, see the WinCALIS Author on-line User's Manual.

paste

To copy the contents of the Clipboard into an application. Many applications have a Paste command on the Edit menu that performs this task.

patch map

The part of a channel-map entry that translates instrument sounds, volume settings, and (optionally) key values for a channel.

path

Specifies the location of a file within the directory tree. For example, to specify the path of a file named SYSTEM.INI located in the WINDOWS directory on drive C, you would type `c:\windows\system.ini`.

PIF (program information file)

A file that provides information about how Windows should run a non-Windows application. For example, you use a PIF to direct Windows to run a non-Windows application in a full screen rather than in a window, or to specify a start-up directory.

pixel

The smallest graphic unit that can be displayed on the screen, usually a single-colored dot. Pixel is the abbreviation for picture element. *Also known as* pel.

plotter font

A font created by a series of dots connected by lines that can be scaled to different sizes (used by plotters). *Also known as* vector font.

point

To move the pointer on the screen until it rests on the item you want to select.

point size

The height of a printed character specified in units called points. A point equals 1/72 of an inch. *Also known as* font size.

pointer

The arrow-shaped cursor on the screen that follows the movement of a mouse (or other pointing device) and indicates which area of the screen will be affected when you press the mouse button. The pointer may change shape during certain tasks. *Also known as* mouse pointer.

pointing device

An input device you use to move the pointer on the screen, choose commands, press buttons, select text, create drawings, and so on. For example, a mouse is a pointing device.

pop-up program

A memory-resident program that is loaded into memory but isn't visible until you press a certain key combination or until a certain event occurs, such as receiving a message. *See also* memory-resident program.

port

A connection or socket on the computer. Ports are used for connecting devices, such as printers, monitors, and modems, to your computer, and for sending information from your computer to the devices. Serial ports (COM) and parallel ports (LPT) are the most commonly used ports.

primary task list

The set of tasks between task "0" and task "end" in a WinCALIS script. These tasks will be executed by WinCALIS in the order in which they appear in the script unless they are deleted or reordered by the WinCALIS author using the <ADD>, <DELETE>, or <INSERT> commands.

print queue

A list of files that have been sent to a printer. The list indicates the file currently printing and those waiting to be printed.

printer driver

A program that controls how your computer and printer interact. A printer-driver file supplies Windows with such information as the printing interface, description of fonts, and features of the installed printer.

printer fonts

Fonts that are stored in your printer's memory, or soft fonts that are sent to your printer before a document is printed. Because the

font works only with the printer, Windows can only approximate the font's appearance on the screen. *See also* screen font, soft font, TrueType fonts.

program file

An executable file that starts an application or program. A program file has an .EXE, .PIF, .COM, or .BAT filename extension.

program items

The applications and documents that make up a group in Program Manager. Program items are represented as icons in a group window.

program-item icon

The icon that represents an application or document in Program Manager. You can start an application or open a document by choosing its program-item icon. (To choose the icon, you double-click it. Or you can use the arrow keys to move the selection cursor to it, and then press ENTER.)

proportional font

A font in which different characters have varying widths, such as Times New Roman. *See also* monospaced font.

protected mode

A computer's operating mode that is capable of addressing extended memory directly.

protocol

A set of rules that define how computers communicate with each other.

pull-down menus

A pull-down menu is a menu item which is accessed by clicking the mouse on the menu bar (The bar at the top of the screen containing various options). Pull-down menus can also be opened by pressing Alt + the key for the letter which is underlined in the name of the pull-down menu. Use the mouse or the Up and Down Arrow keys to highlight a selection within a pull-down menu. Press ENTER or double-click the mouse to execute your selection.

Question Window

The main WinCALIS window in which the student views information and enters a response to a question or command. By default the question window fills any space not taken by other windows.

quick format

Deletes the file allocation table and root directory of a disk, making it appear empty, but it does not delete the file information on the disk or scan the disk for bad areas. This serves as a quick way to format a previously formatted disk.

Q

Quicktime

A digital video standard originally created for the Apple Macintosh computer. A version with Windows MCI support has been created, called "Quicktime for Windows." It is based on the

Macintosh Quicktime digital video standard. A Quicktime for Windows Runtime player is available from Apple Corporation.

R

RAM (random access memory)

The memory that can be used by applications to perform necessary tasks while the computer is on. When you turn the computer off, all information in RAM is lost. *See also* virtual memory, ROM.

RAM drive

A portion of memory that is used as if it were a hard disk drive. RAM drives are much faster than hard disks because your computer can read information faster from memory than from a hard disk. However, information on a RAM drive is lost when you turn off or restart your computer. *Also known as* virtual drive.

read-only file

A file that you can read but cannot make changes to. The read-only attribute specifies whether a file is read-only. *See also* attributes.

reduce

To minimize a window to an icon at the bottom of the desktop by using the Minimize button or the Minimize command. A minimized application continues running, and you can select the icon to make it the active application.

resolution

Indicates the number of dots that make up an image on a screen or printer. The larger the number of dots, and thus the higher the resolution, the finer and smoother images can appear when displayed at a given size.

Restore button

The small button containing both an up and down arrow at the right of the title bar. The Restore button appears only after you have enlarged a window to its maximum size. Mouse users can click the Restore button to return the window to its previous size. Keyboard users can use the Restore command on the Control menu.

ROM (read-only memory)

Memory that can be read, but not modified. The contents of this memory is retained even when the computer is turned off.

root directory

The top-level directory of a disk. The root directory is created when you format the disk. From the root directory, you can create files and other directories.

S

saturation

In Control Panel, the purity of a color's hue, moving from gray to the pure color. For example, a very dark blue may have mostly black with only a small amount of blue, and thus would have a very low saturation. *See also* hue, luminosity.

scalable typeface

A set of letters, numbers, punctuation marks, and symbols that are a given design but can be scaled to any size.

scaled point size

A point size that approximates a specified point size for use on the screen. For example, text that prints at 10 point on the printer may be represented by a slightly larger font on the screen to make up for the screen's lower resolution.

scan exercise

A common type of exercise in which the student is required to read a text and then answer questions about it.

Scratchpad Window

The WinCALIS Author Scratchpad Window is a Unicode Notepad, which is built into the Author program. It works much like a plain-text editor like the Windows Notepad or the DOS Editor, but uses Unicode exclusively.

The Scratchpad window differs from the other Author windows. When you save a document file in Scratchpad, WinCALIS does not generate a *.WCA file. Without this file, WinCALIS Author does not know which Question Window pertains to which anticipated response. Consequently, manually written scripts or scripts modified in the Scratchpad can be understood by WinCALIS, but cannot be interpreted by WinCALIS Author. In other words, modifications made in the Scratchpad will not be visible in WinCALIS Author.

screen font

The font that is displayed on your screen, which may approximate a printer font so that documents look approximately the same on the screen as they do when printed. *Also known as* raster font or bitmap font.

screen saver

A moving picture or pattern that appears on your screen when you have not moved the mouse or pressed a key in Windows for a specified period of time. Screen savers prevent screen damage that is caused when the same areas of light and dark are displayed for long periods of time.

script

(1) A WinCALIS exercise, coded in the CALIS authoring language. A script consists of one or more tasks. (2) A character set or alphabet, as in "Roman script," "Arabic script," "Cyrillic script," etc.

scroll

To move through text or graphics (up, down, left, or right) in order to see parts of the file or list that cannot fit on the screen. The cursor control keys or numeric keypad are used to scroll text in text-based CALIS. Either the mouse or keyboard keys (e.g., Shift+Up Arrow and Shift+Down Arrow to scroll the Text Window) can be used in WinCALIS.

scroll arrow

An arrow on either end of a scroll bar that you use to scroll through the contents of the window or list box.

scroll bar

A bar that appears at the bottom and/or right edge of a window whose contents are not entirely visible. Each scroll bar contains a scroll box and two scroll arrows.

scroll box

In a scroll bar, the small box that shows the position of information currently in the window or list box relative to the contents of the entire window.

secondary task list

The set of tasks in a WinCALIS script which will be executed by WinCALIS after completing the primary task list. It is constructed by using <ADD> commands and is executed in the order in which they have been ADDED, unless they are deleted or reordered by the WinCALIS lesson author using the <ADD>, <DELETE>, or <INSERT> commands.

select

To mark an item so that a subsequent action can be carried out on that item. You usually select an item by clicking it with a mouse or pressing a key. After selecting an item, you choose the action that you want to affect the item. *See also* choose.

selection cursor

The marking device that shows where you are in a window, menu, or dialog box and what you have selected. The selection cursor can appear as a highlight or as a dotted rectangle around text.

separator

In the CALIS authoring language, a semicolon (;), called a "separator," is used in anticipated response statements to separate the student answer portion from the feedback portion.

serial interface

An interface between a computer and a serial device, such as a printer or modem, by which the computer sends single bits of information to the device, one after the other. Serial, asynchronous, and RS232 interfaces are all the same type. *See also* parallel interface.

serial port

A connection on a computer, usually COM1, where you plug in the cable for a serial device. Common serial devices are printers and modems. Windows supports COM1 through COM4.

serial printer

A printer using a serial interface, which you connect to a serial port.

server

A computer that provides disk space, printers, or other services to computers over a network. *See also* network drive, network printer.

server application

A Windows application that creates objects that can be linked and embedded into other documents. *See also* client application.

shortcut key

A key or key combination that you press to carry out a command or action. If a command has a shortcut key, the key combination is listed to the right of the command name on the menu. For example, pressing the shortcut key Alt+F4 closes the active application.

simple device

An MCI device that does not use media files. For example, an audio compact-disc player is a simple device. *See also* compound device, Media Control Interface.

soft font

A font, usually provided by a font vendor, that must be installed on your computer and sent to the printer before it can be printed. *Also known as* downloadable font.

software

The set of instructions that make computer hardware perform tasks. Programs, operating systems, device drivers, and applications are all software.

solid color

A color that appears on a display when all pixels (pels) are the same color. On a monochrome display, there are only two solid colors: black and white. *See also* nonsolid color.

sound driver

A program that applications use to play sounds on your computer.

sound file

A file containing audio information that an application can play on your computer. You can assign a sound file to a system event, such as starting Windows, so that you are alerted by a sound when the event occurs.

source directory

The directory that contains the file or files you intend to copy or move.

source document

The document from which a linked object originates.

special characters

Characters not found on your computer's keyboard. These are accessible through Character Map, an application in the Accessories group.

special response

A student response other than keyboard input followed by pressing ENTER. Examples include simply pressing ENTER, using the plus and minus keys to pass among tasks, and using the mouse to click on a portion of the screen.

split bar

Divides a window into two parts. In File Manager, a directory window is divided by a split bar: The directory tree is on the left, and the contents of the current directory are on the right.

standard mode

A Windows operating mode that can be used with 80286, 80386, and 80486 computers. This mode provides access to extended memory and also enables you to switch between non-Windows applications, but it does not provide virtual memory or enable non-Windows applications to run in the background or in a window. *See also* 386 enhanced mode.

status bar

A line of information usually located at the bottom of a window. In File Manager, the status bar shows the number of bytes available on the disk and the total disk capacity. Not all windows have a status bar, and in many applications, it is optional.

subdirectory

A directory within another directory.

swap file

An area of your hard disk that is set aside for exclusive use by Windows in 386 enhanced mode. Windows temporarily transfers information from memory to the swap file to free memory for other information. Swap files can be either temporary or permanent. *See also* application swap file.

synthesizer

A device that produces sound from digital instructions rather than from recorded sound.

system disk

A disk that contains the MS-DOS system files necessary to start MS-DOS. You can have more than one system disk; however, the disk from which MS-DOS is loaded when you start your computer is called the boot disk.

system file

A file that contains information required for running the MS-DOS operating system. A system file is not shown in the directory listing. The system attribute specifies whether a file is a system file. *See also* attributes.

system time

The time according to your computer's internal clock.

SYSTEM.INI file

A Windows initialization file that contains settings you can use to customize Windows for your system's hardware.

T

task

The basic building block of a WinCALIS script. It is a set of WinCALIS commands which will be executed during the WinCALIS lesson. Most commonly, tasks require student input.

task label

An arbitrary name given to a WinCALIS task consisting of a string of characters. A task label may be any size, but WinCALIS will ignore more than fifteen characters.

task list

(1) In WinCALIS, the list of all tasks (questions) to be presented to the student in sequential order. *See also* primary task list, secondary task list. (2) In general in Windows, a window that shows all the applications you have running and enables you to switch between them. You can open an application's Task List by choosing Switch To from the Control Menu or by pressing Ctrl+Esc.

task options

WinCALIS settings which define the behavior of a WinCALIS script, either globally or for a particular task within a WinCALIS script.

taskbase

A text CALIS (CALIS for DOS) utility for preparing one CALIS script for use within another.

terminal emulation

In communications, a setting that causes your computer to emulate a hardware terminal (a device used to display data received from a remote computer).

text box

In a dialog box, a box in which you type information needed to carry out a command. The text box may be blank or may contain text when the dialog box opens.

text editor

An application used to create, view, and modify text files; for example, Notepad.

text file

A file containing only letters, digits, and symbols. A text file usually consists of characters coded from the ASCII character set, but in WinCALIS usage, a text file is a plain-text file coded in the Unicode character set. *Also known as* data file. *See also* document file, binary file.

text formatting commands

WinCALIS commands which change the appearance of text to the student. They do not affect the order of execution of a lesson.

text mode

The mode that enables applications to display text but not graphic images. Many non-Windows applications run in text mode. *See also* graphics mode.

Text Window

A WinCALIS window which is generally used to show text or instructions to the student for a number of tasks. By default, it appears above the question window.

text transfer

A method for transferring text files from Terminal to a remote computer.

tile

A way of arranging open windows so that no windows overlap but all windows are visible. Each window takes up a portion of the screen.

timeout

If a device is not performing a task, the amount of time a computer should wait before detecting it as an error.

time slice

The amount of processor time allocated to an application, usually measured in milliseconds. Each application can run for this

specified amount of time before the next application is given control.

title bar

The horizontal bar (at the top of a window) that contains the title of the window or dialog box. On many windows, the title bar also contains the Control-menu box and Maximize and Minimize buttons.

TrueType fonts

Fonts that are scalable and sometimes generated as bitmaps or soft fonts, depending on the capabilities of your printer. TrueType fonts can be sized to any height, and they print exactly as they appear on the screen. *See also* printer font, screen font.

TSR

Acronym for terminate-and-stay-resident program. *See* memory-resident program.

Tutorial Window

A WinCALIS window which is opened by pressing Shift + F1 - F10 or selecting a tutorial window from the "Tutorials" menu on the WinCALIS menu bar. Tutorial windows may "pop up" during a lesson as well.

The student can resize and keep these windows on-screen while working a lesson. The "Tutorials" Pull-down menu option will list all available tutorial windows within a lesson. Tutorial windows can also be shown at specific times as specified by the author using the <POPUP> command.

Tutorial windows are either written into a script using the <Fx> command or are separate text files which are brought into a script using the <IMPORT> command. Tutorial windows can contain WinCALIS text formatting commands but may not contain dynamic commands such as <DELAY>, <IMPORT>, and <DELETE>.

U

Unicode character set

A character set consisting of potentially up to 65,536 fixed-width 16-bit codes, uniquely identifying virtually all the characters used by the languages of the modern world. It is registered as a subset of the ISO (International Standards Organization) 10646 32-bit international coding standard. Unicode is used exclusively by WinCALIS 2.x to represent all characters, whether Roman or non-Roman.

The least significant 7 bits of Unicode are identical with the ASCII (the American Standard Code for Information Interchange) standard of 128 character codes. The least significant 8 bits of Unicode are identical with the ANSI (American National Standards Institute) standard of 256 character codes, which is used by the U.S. version of Microsoft Windows. ANSI is also known as "Latin 1" and is cross-registered as ISO 8859-1 and as ECMA-94 by the European Computer Manufacturers Association.

Unicode text file

A plain-text file consisting of only valid 16-bit (double-byte) characters found in the standard Unicode character set and containing no formatting commands or other non-Unicode characters, as found in many word processor-formatted documents.

UniEdit

The WinCALIS Multilingual Text Editor program. UniEdit is identical in editing function with the WinCALIS Author program,

except that it has no automatic script authoring capability. It is a Unicode-compliant Windows Notepad.

up directory icon

This symbol:

[..]

moves you up one level in the DOS directory structure of the current drive when you double-click on it. It appears in all file list boxes.

upper ASCII

The extended character set above ASCII value 127, defined by IBM Corporation for use in the U.S. version of the operating system DOS on the IBM PC and compatibles. Often referred to within Windows as the OEM character set. This set contains accented European characters and mathematical symbols. In Text CALIS and WinCALIS 1.x it is replaced with foreign character sets for use with non-ASCII alphabets. WinCALIS 2.x is Unicode-compliant and no longer uses "upper ASCII" and no longer substitutes its own arbitrary character codes.

upper memory area

The 384K area of address space adjacent to the 640K of conventional memory. This area is usually reserved for running your system's hardware, such as your monitor, and is not considered part of total memory because applications cannot store information in this area. In 386 enhanced mode, Windows can access unused portions of this area.

upper memory blocks (UMBs)

Areas of the upper memory area that contain general-purpose memory and that can be used to hold device drivers or other memory-resident programs in order to leave more conventional memory available for applications.

V

Video for Windows

The name of Microsoft's device for playing digital video. The Video for Windows device supports video sequences made from 8-bit, 16-bit, and 24-bit color images, with or without an accompanying sound track. Video for Windows plays digital video files with the file extension *.AVI (standing for "audio-video-interleaved," which describes how audio and video are stored within the *.AVI file).

Although Windows 3.1 did not originally provide built-in MCI support for digital full-motion video, Microsoft issued "Video for Windows" as a separate enhancement package in November 1992. It has since been upgraded to version 1.1 and is integrated in forthcoming versions of Windows (Windows 95).

video overlay card

A PC-compatible expansion card which allows real time processing of an NTSC video signal into a digitized signal which can be sized and displayed on a VGA monitor.

virtual memory

A memory-management system used by Windows in 386 enhanced mode that enables Windows to run as if there were more memory than is actually present. The amount of virtual memory available equals the amount of free random-access memory (RAM) plus the amount of disk space allocated to a swap file that Windows uses to simulate additional RAM.

volume label

A name that identifies a disk. The volume label appears in the title bar of a directory window in File Manager.

W

wallpaper

An image being displayed on the desktop background.

wild card

A character that represents one or more other characters. In DOS filenames, you can use the question mark (?) to represent any single character and the asterisk (*) to represent any character or group of characters that might match that position in other filenames. For example *.EXE represents all files that end with the .EXE filename extension.

In the CALIS Authoring Language, you can use the caret (^) to represent any single character, the asterisk (*) to represent any character or group of characters bounded by spaces, and the ampersand (&) to represent any character or group of characters, including the space character.

WIN.INI file

A Windows initialization file that contains settings you can use to customize your Windows environment.

WinCALIS Author

A WinCALIS program which enables the user to create various types of WinCALIS lessons quickly and easily without knowing WinCALIS commands and syntax.

WINCALIS.WGI file

The WinCALIS global information table file. It contains information necessary for WinCALIS to check or ignore upper and lower case relations, punctuation, and accented characters in student answers, in any language, regardless of the currently selected language keyboard. It is automatically generated from the editable ASCII file WINCALIS.INF. For details on modifying the global information file, see "Configuring and Customizing WinCALIS Author" in On-line Help.

window

A rectangular area on your screen in which you view an application or document. You can open, close, and move windows, and change the size of most windows. You can open several windows at a time, and you can often reduce a window to an icon or enlarge it to fill the entire desktop. Sometimes windows are displayed within other windows. *See also* application window, document window.

Windows application

"Windows application" is used in this guide as a shorthand term to refer to an application that is designed to run with Windows and does not run without Windows. All Windows applications follow similar conventions for arrangement of menus, style of dialog boxes, and keyboard and mouse use.

Windows character set

The character set used to display Windows and Windows applications. It is based on the ANSI character set. *See also* ANSI character set.

word wrap

A feature that moves text from the end of a line to the beginning of a new line as you type. With word wrap, you do not need to press ENTER at the end of each line in a paragraph.

workspace

The area of a window that displays the information contained in the application or document you are working with.

X Y Z***XMS memory***

Extended memory that can be accessed by using the eXtended Memory Specification from Lotus, Intel, Microsoft, and AST. Applications work with an extended-memory manager, which makes sure only one application is using a portion of memory at any one time.

Index of Figures

Figure 1: WinCALIS Opening Screen	10
Figure 2: Keyboard Help	14
Figure 3: Putting the Keys Together.....	18
Figure 4: Selecting Multiple Choice Question Type	22
Figure 5: Keyboard Help image.....	49
Figure 7: Insert CALIScript Command Tool	58
Figure 6: Response Dialog Box.....	67
Figure 7: Neutral Re-Execute as a 'clue' task	68
Figure 8: Cloze Exercise with default settings.....	74
Figure 9 Cloze Exercise with options:	75
Figure 10: All <i>n</i> Answers Example	80
Figure 11: Feedback appears in the D window.....	81
Figure 12: Tab Stop in Multiple Choice	82
Figure 13: Indicating a hotspot on a graphic <HOTSPOT 20, 30, 40, 60>	96
Figure 14: Question as originally presented.....	199
Figure 15: After a Give Up response, sentence is rewritten through feedback.....	200
Figure 16: Indicating a hotspot on a graphic <HOTSPOT 20, 30, 40, 60>	201
Figure 17: Sample Illustrations for Courseware Graphics.....	231
Figure 18: Multimedia Editor Controls	236
Figure 19: To play a clip (section of a multimedia file)	238
Figure 20: Insert Audio Clip Dialog Box	242
Figure 21: Choosing "Record Student" in the Audio Clip dialog box	246
Figure 22: MCI Control Bar / Media Controls	247
Figure 23: On-line Dictionary Box	256
Figure 24: Sample Korean - English Dictionary Entry	260
Figure 25: Example Dictionary Entry, using "Also"	261
Figure 26: Sample path.db file	266
Figure 27: Sample studrec.db file	266
Figure 28: Sample of a WinCALIS Level 1 records file.....	267
Figure 29: Sample of a WinCALIS Level 2 records file.....	267
Figure 30: Choose 'startup.fsl' for the forms that view records tables	269
Figure 31: Viewing WinCALIS records tables through Paradox form	270
Figure 32: Chinese Status Bar	297
Figure 33: Hiragana keyboard.....	304
Figure 34: Korean Phonetic keyboard (unshifted).....	314
Figure 35: Korean Standard keyboard (unshifted)	315
Figure 36: Korean Kong keyboard (unshifted)	315
Figure 37: Korean Morse keyboard (unshifted).....	316
Figure 38: Korean Hanja by shape keyboard.....	317
Figure 39: Korean Transliteration keyboard.....	317

Index

- *
 - asterisk indicates unsaved material..... 63
- <
 - < >
 - appearing in scripts..... 111
- A**
 - ACCENT 213
 - checking for correct placement of accents..... 214
 - Accents and Diacritics 17
 - accent keys 17
 - the overstrike key 17
 - ADD 188, 390
 - ALL 214, 215
 - All Answers option 78
 - All n Answers option 80
 - ALL:n 216
 - Allow Giving Up option 80
 - Allow Passing option 78
 - Allow Repeat option 79
 - Amharic
 - keyboard layout 367
 - angled brackets
 - appearing in scripts..... 111
 - Animation**
 - creating clips** 237
 - inserting into lessons**..... 242
 - preparing files**..... 226
 - WinCALIS support..... 229
 - Answer Evaluation **213**
 - Anticipated Response/Feedback Dialog Box **66**
 - ANY 214
 - APPEND
 - example script containing **283**
 - APPEND (as used with EXPORT)..... **284**
 - Applications**
 - executing outside applications** 183
 - Arabic
 - keyboard layout 367
 - Scripts reference 292
 - Armenian
 - keyboard layout 368
 - ASSIGN
 - using with variables 278
 - asterisk
 - indicates unsaved material..... 63
 - Attribute 103
 - Font 52
 - Audio**
 - *.WAV files..... 228
 - creating clips** 237
 - incorporating** 180
 - inserting into lessons**..... 241
 - multimedia option..... 247
 - options 245
 - options and commands 248
 - preparing files..... 226
 - recording..... 226
 - special options 246
 - AUDIO
 - special considerations..... 196
 - Author..... 218
 - B**
 - BEEP 178
 - sounding a..... 109
 - special considerations..... 196
 - Belarusian
 - keyboard layout 370
 - BLA..... *See* BLANK
 - Blank
 - [Window] option 82
 - BLANK 127, 164
 - special considerations..... 196
 - to clear the Interactive (Question) Window..... 210
 - Blank line
 - tip for removing in Cloze..... 72
 - Blank Window
 - format command..... 50
 - blanks**..... *See* <FILL>
 - generating and sizing**..... 174
 - BMP *See* Graphics
 - Inserting *.BMP graphic file into lesson 56
 - BOLD 103, 141, 152
 - Bopomofo (Chinese) 298
 - keyboard layout 372
 - Bulgarian
 - keyboard layout 370
 - BUTTON 94, 197
 - action 197
 - Creating 277
 - example script containing 197
 - types and options 277
 - Buttons
 - ANSI text..... 277
 - special response..... 94
 - Unicode text 277
 - C**
 - CAL
 - and Script *.WCA files..... 38

as Document file	40	Background	155
Editor limits with *.CAL file	33	commands	153
File Structure	136	Commands for text	153
loading *.CAL file into Scratchpad	137	Foreground	154
modifying	39	commands	103, 105
note to users of text-based CALIS	2	COMMENT	285
Text Conversion	9	Composition Editor	
CAL (*.CAL)	321, 343, 347	and Look Up Word	15
CALIScript		explained	16
explained	135	Conversion	
Inserting from Tools menu item	57	code	252, 347
CALIScript language		Converting	
in relationship to WinCALIS	135	*.CAL files	137
Cangjie		Coordinates	172
keyboard layout	372	Copyright Information	ii
Cangjie Input Method	298	CORRECT	176
CAPITAL	216	displaying in feedback	112
CCCII	350, 355	displaying in windows other than feedback	112
CE	<i>See</i> CENTER	formatting with spaces	177
video location center portion	181	parameters	113
Center		Creating a lesson	19
format command	50	Entering text	20
CENTER	107, 157	Giving directions	20
Center option	78	Producing Tasks	
Center Text in the Window option	83	<i>Cloze Exercise</i>	26
Character-to-Font Mapping File WCFNT.INF	328	Multiple Choice	22
Check Accents option	78	True/False	24
Check Case		Producing Tutorials	129
option	77	Creating a Task Database	191
Check Punctuation option	78	GLUE	193
Check Score option	78	RANDOM	191
Chinese	293	UNGLUE	195
Function Keys	295	Cursor	
status bar	296	positioning at specific <MARK>s	172
CLEAR	127, 128, 129, 173	positioning at specific MARKs	116
example script containing	127, 173	positioning in windows	104, 170
clips		CURSOR	104, 105, 132, 172
creating	237	example script containing	170
inserting into lessons	241	D	
naming	238	D	
options	245	example script containing	124
saving	239	use of D window	210
CLOSE	285	D Window	160
Cloze exercise		overview	160
clues	73	sending feedback to	210
fixed width blanks	291	D Window (Diagnostic Feedback Window)	162
hints with Japanese language	313	Default statements	142
options	73	Defaults	
producing	25	task	207
Cloze Exercise		DELAY	286
under the Task menu item	54	execution	108
Code Conversion	252, 325, 347, 356	OFF	159
Eight-bit	347	special considerations	196
Sixteen-bit	350	using with n	108
Twenty-four-bit	355	using with n for a time delay	159
Color	100	DELETE	186
available in WinCALIS	100		

-
- delimiters 138, 360
 - Diagnostic Feedback Window
 - activating 162
 - feedback placed here 147
 - formatting 102
 - using for feedback 81
 - Dictionaries
 - User-Supplied 254
 - Dictionary
 - How to create your own 262
 - Japanese-and-English 258
 - Korean-and-English 259
 - options 256
 - Directions
 - incorporating 65
 - Directions window 243
 - DISABLE 174
 - buttons 277
 - Disable Keyboard option 81
 - DONE 284
 - dragdrop
 - example script containing 197
 - DRAGDROP 197
 - DRAW 286
 - E**
 - EACC 355, 356
 - Edit Menu 44
 - Editor
 - in Author, explained 31
 - in Author, limits 33
 - Multimedia 235
 - EIGHTH 181
 - ENABLE 174
 - buttons 277
 - English/W.Europe
 - keyboard layout 367
 - ENTER 94, 198
 - as a Special Response 198
 - example script containing 203
 - Enter Answer Length n: option 81
 - Enter Number of Answers n: option 80
 - EXECUTE**
 - executing outside applications** 183
 - F**
 - Farsi xii, 49
 - Features
 - new 289
 - Feedback
 - directing to Windows 113
 - refining 111
 - File Menu 38
 - files
 - outside of WinCALIS 129
 - FILL 123, 174
 - example script containing 124
 - FLASH
 - a BUTTON 277
 - Flash Card 70
 - font
 - indicating default font 285
 - Font
 - format command 50
 - fonts
 - adding 325
 - and printing 273
 - considerations when installing to a network 343
 - files 331
 - FONTSIZE 105, 156
 - example script containing 124, 157
 - Form/Symbols
 - keyboard layout 372
 - Format Menu 50
 - French
 - keyboard layout 368
 - FULL 181
 - G**
 - Generate Index 264
 - Georgian
 - keyboard layout 368
 - German
 - keyboard layout 369
 - GIVE 217
 - setting number of attempts before giveup 217
 - GIVE: n 217
 - GIVEUP 199
 - GIVEUP (options) 217
 - GIVEUP (special response) 95
 - Giveup repeats task at end option 78
 - Glossary 377
 - GLUE 192, 193
 - Graphics** 229
 - command strings inserted by Author 233
 - creating 229
 - Graphic Dialog Box 232
 - incorporating into lessons** 230
 - requirements** 229
 - saving 229
 - sizing windows correctly for 233
 - sources 231
 - student answer line, conflict with 233
 - subdirectory 234
 - Greek-Phonetic
 - keyboard layout 369
 - GROUP:grouptask [,exitgrouptask 218
 - GROUPBOX 277
 - GROUPLOCK 284
 - task option 220

H

Hangul (Kong)	
keyboard layout.....	374
Hangul (Phonetic)	
keyboard layout.....	374
Hangul (Standard)	
keyboard layout.....	374
Hanja	257
Hebrew	
Scripts reference.....	292
Hebrew-Israel	
keyboard layout.....	373
Hebrew-Phonetic	
keyboard layout.....	373
Help.....	64
Help Menu	9, 63
Help Menu (WinCALIS).....	17
Hot Text	
creating with MARKs	118
HOTSPOT	
Creating.....	200

I

IF	
using with variables.....	282
IMPORT	131, 167
example script containing.....	131, 168
exceptions/restrictions.....	168
Importing	
ASCII files into Scratchpad	138
files.....	41, 167
Incorporating Graphics	179
Info Line in Task Windows option	83
INI	
WINCALIS.INI	319
INSERT	
tasks	187
tasks randomly	191
Inserting	
clips or graphics in WinCALIS.....	55
Install	
demonstration lesson.....	8
Optional files.....	8
Installation	7
Interactive	66
Interactive (Question)	35
Interactive (Question) window	
methodology.....	66
Response types.....	66
Introduction.....	xi
Italian	
keyboard layout.....	369
ITALIC	152
set attribute.....	288

J

Japanese	304
-and-English Dictionary	258
Hints for authors.....	313
Kana Chart	308
Kanji by Shape.....	304
Spelling Variation	310
Typing Peculiarities	305
JIS Hiragana	
keyboard layout.....	373
JIS Katakana	
keyboard layout.....	374

K

Kana Chart	308
Kangxi Radical Chart.....	301
Katakana	306
KEY	202
as a Special Response	287
designating mouse or keyboard input.....	212
keyboard shortcuts	30, 49
Korean.....	314
-and-English Dictionary	259

L

L	
default font statement.....	328
to indicate default font	285
Language	
Options.....	14, 15, 59
Right-to-Left languages	292
Language Keyboard	
defining files	330
INI Files	331
Language Menu.....	48
Languages	
supported by WinCALIS.....	1
LAST	
displaying a student's last response	176
displaying student's last response in feedback..	111
LASTMARK	
displaying last text marked by student	287
LC	
video location lower center	181
Left margin Column Number option.....	83
LENGTH	
limiting length of student's answer	212
Lesson	
How to run	84
Lesson Description Window	34
Lesson Menu	
WinCalis Menus.....	11
Lesson Presentation.....	100
Limit Answer Length option	81
LL	
video location lower left.....	181

-
- LOC
 - video location 181
 - Location
 - multimedia option 247
 - Look Up Word 254
 - LR
 - video location lower right 181
 - M**
 - Margin
 - left margin column number 83
 - Margins
 - print settings 273
 - MARK 116
 - Mark Spelling option 78
 - mark\$ *See* Variables, global
 - Marks
 - and hot text 119
 - Max. Tries option 79
 - MAXTRIES
 - setting maximum number of attempts 217
 - MCI Control Bar 247
 - Show/Hide 287
 - Media Commands 178
 - Media Controls 247
 - Menus
 - in WinCALIS Author 38
 - Min. Tries option 79
 - MMD files
 - creating 228
 - opening 237, 239
 - MMD Files 228
 - MOUSE
 - appearance of the hand pointer 213
 - as a Special Response 205
 - in combination with MARK 118
 - special response 97
 - Multilingual Printer Program 273
 - Multimedia
 - *.MMD files 228
 - animation 225
 - audio 226
 - designing lessons 225
 - function in lessons 224
 - graphics 229
 - requirements 223
 - Multimedia Editor
 - creating clips** 237
 - function of 58
 - naming clips** 238
 - opening *.MMD files 237, 239
 - saving clips 239
 - Multiple Choice
 - explained 71
 - formatting questions 291
 - producing question 22
 - under the Task menu item 53
 - with randomize option 79
 - N**
 - N
 - displaying the task sequence number 177
 - New
 - features 289
 - No Mouse Hand Pointer option 81
 - NOACCENT *See* ACCENT
 - NOBLANK *See* BLANK
 - NOCAPITAL *See* CAPITAL
 - NOD *See* D
 - NOGIVEUP *See* GIVEUP
 - NOKEY *See* KEY
 - NOLENGTH *See* LENGTH
 - NOMOUSE *See* MOUSE
 - NOPASS *See* PASS
 - NOPRAISE *See* PRAISE
 - NOPUNCT *See* PUNCT
 - NOREPEAT *See* REPEAT
 - NORMAL
 - in BUTTON creation 277
 - set attribute 288
 - NOSCORE *See* SCORE
 - NOSHOW *See* SHOW
 - O**
 - Opening
 - document file (contents of one window) 41
 - WCA Script files 39
 - Opening Screen 8
 - Options
 - ACCENT 214
 - ALL 214
 - ALL:n 216
 - ANY 214
 - CAPITAL 216
 - Cloze exercise 73
 - for script generation 83
 - GIVE:n 217
 - GIVEUP 95, 199
 - GROUP
 - grouptask 218
 - KEY 212
 - language 59
 - LENGTH 212
 - LENGTH:n 212
 - MAXTRIES:n 217
 - More 79
 - MOUSE 213
 - NO ACCENT 214
 - NOBLANK 210
 - NOCAPITAL 216
 - NOD 210
 - NOGIVEUP 217

NOKEY	212	the contents of a window	285
NOLENGTH	212	Print settings	
NOMOUSE	213	Multilingual Printer (MLP.EXE)	273
NOPASS	208	Print Settings	276
NOPRAISE	211	TrueType Printer	274
NOPUNCT	216	Printing	273
NOREPEAT	208	*.CAL files	274, 275
NOSCORE	209	Multilingual Printer program (MLP.EXE)	273
NOSHOW	211	TrueType Printer program (MLPTT.EXE)	273
OR	98	Producing Tasks	
PASS	208	cloze	25
PRAISE	210	short Answer	24
PUNCT	216	PUNCT	
QUIZ	221	requiring correct placement of punctuation	216
REPEAT	208	task option	216
SCORE	209	Q	
script	52	QUARTER	181
script and task	77	Question	
SHOW	211	sequencing in WinCALIS	76
video and animation	246	Question Window	66. <i>See</i> Interactive (Question)
Options:D	210	Window	
OR		methodology	66
special response	98	Quiz	82
with special responses	94	QUIZ	221, 285
OWNERDRAW		R	
in button creation	277	RADIOBUTTON	277
in BUTTON creation	277	RANDOM	191
P		Random Task Order option	83
Pass		Randomize Multiple Choice option	79
between tasks in WinCALIS	13	Registration	3
PASS		REPEAT	208
option	208	REPORT	284
special response	99	Requirements	2
using special responses	99	response\$	<i>See</i> Variables, global
PASS [n]	287	result	<i>See</i> Variables, global
Pass Between Groups option	80	REVERSE	152
Pass Through Window	36	Right-to-Left Scripts	292
PASSBACK	206	running a lesson	84
special response	100	Russian-CALIS	
using special responses	100	keyboard layout	371
PASSBACK [n]	287	Russian-Standard	
Persian-CALIS		keyboard layout	371
keyboard layout	368	S	
Pinyin	293	Save As...	
Play		document or window contents (e.g.*.CAL)	43
multimedia option	247	script (*.WCA)	40
PLAY		Saving files	
video or audio	180	*.CAL files	42
POPHELP	184	*.WCA files	39
POPUP	130	score	78
"Popping Up" Tutorials	130	Score	
Popup (Directions)	34	weighted score explained	209
PRAISE	210	SCORE	209, 210
Praise option	78	task-lesson interaction	208
Presentation (Text) Window	35		
PRINT			

-
- Scratchpad
 - opening a *.CAL file 137
 - Scratchpad Window 35
 - Script
 - How to run 84
 - Your First 19
 - Script [* .wca] 38
 - Script Generation Options 83
 - Script Menu 52
 - Script Options 77
 - Serbian
 - keyboard layout 371
 - SET
 - ANSWER length 174
 - ATTRIBUTE 287
 - color in Lesson Description Window 102
 - colors, background 155
 - colors, foreground 154
 - colors, foreground and background 101
 - defining numeric values** 174
 - DISABLE ENABLE Tutorial Windows 175
 - FILL 175
 - font sizes 157
 - fontsize 106
 - fontsize in Lesson Description Window 106
 - last length 113
 - numeric values for use with LAST and CORRECT 113
 - Setup
 - installation 7
 - Short Answer
 - under the Task menu item 53
 - SHOW 211
 - MCI control bar 287
 - Show Correct option 78
 - SIXTEENTH 181
 - Size
 - multimedia option 246
 - SIZE
 - in WinCALIS lessons 111
 - video 181
 - Sizing Text 105
 - Sizing Windows 109
 - Sôketsu
 - (Cangjie Input Method) 298
 - special responses
 - ENTER 94
 - GIVEUP 95
 - Special Responses 94, 196
 - <PASS>
 - example script containing 206
 - <PASSBACK>
 - example script containing 207
 - BUTTON 94, 197
 - conditions for general use 151
 - ENTER 94, 198
 - example script containing 198
 - GIVEUP 95, 199
 - HOTSPOT 96, 200
 - IF 96
 - KEY 96, 202
 - example script containing 202
 - MOUSE 97, 205
 - example script containing 205
 - OR 98, 205
 - PASS 99, 206
 - PASSBACK 100, 206
 - SPELLCHECK *See* SPELLMARK
 - Spellchecker
 - link to 327
 - SPELLMARK 119, 181
 - example of 120
 - example script containing 121
 - STATE
 - keyword 277
 - Statement Markers
 - (incorrect answer statement)
 - example script containing 146
 - ! (Directions statement)** 143
 - # (task label statement)
 - example script containing 143
 - \$ (neutral no re-execute response statement)** 146
 - & (spillover statement)
 - example script containing 147
 - / (presentation [text] statement)** 144
 - / (text statement)
 - example script containing 144
 - ? (interactive [question] statement)** 144
 - + (anticipated correct statement)
 - example script containing 145
 - = (neutral re-execute response statement)** 146
 - C (comment statement)
 - example script using 141
 - listed 139
 - Status Bar
 - Chinese 296
 - STRIKEOUT 152
 - Student Chooses # of Tasks option 83
 - SUBST
 - Substitute value of variable 278
 - Support, WinCALIS Technical 4
 - System Requirements 2, 7
 - T**
 - TAB 158
 - example script containing 158
 - to align text on screen 107
 - Tab Stop
 - format command 50
 - in Multiple Choice tasks 82

Task		
adding	188	
Cloze exercise		
explained.....	72	
deleting	186	
Flash Card.....	70	
inserting	187	
Multiple Choice, explained.....	71	
sequencing in WinCALIS.....	76	
Short Answer, explained.....	70	
True/False, explained.....	71	
task list		
defined.....	139	
manipulation.....	185	
primary, defined.....	186	
sample and diagram.....	189	
secondary, defined.....	186	
Task Menu.....	53	
Task Options.....	207	
ACCENT (checking for placement of accents).....	214	
ANY (anticipating responses).....	214	
BLANK.....	210	
CAPITAL (checking for capitalization).....	216	
conditions for general use.....	151	
D window.....	210	
GIVE (setting minimum attempt number)	217	
GIVEUP (allowing giving up)	217	
KEY (designating mouse or keyboard input).....	212	
LENGTH (limiting length of answer).....	212	
MAXTRIES (setting maximum number of attempts)	217	
MOUSE (disabling hand pointer).....	213	
PASS.....	209	
PRAISE (Correct/Incorrect messages in Diagnostic Feedback Window).....	210	
PUNCT (checking for punctuation).....	216	
SCORE.....	210	
SHOW (displaying answer upon giveup)	211	
Tasks		
Random Task Order option.....	83	
TC		
video location top center.....	181	
Text [*.CAL]		
commands		
Save and Save As.....	132	
file structure.....	136	
opening file in Scratchpad.....	137	
reading and writing files.....	137	
sample script.....	136	
Text Attributes.....	103	
CENTER.....	107	
FONTSIZE.....	105	
in Opening Screen.....	9	
TAB.....	107	
aligning text.....	108	
Text Window.....	35	
creating.....	20	
Thai-Chiangmai		
keyboard layout.....	375	
TL		
video location top left.....	181	
Tools		
Insert Button.....	57	
Insert CALIScript Command.....	57	
Insert Graphic.....	56	
Tools Menu.....	55	
Top Line option.....	82	
TOPLINE.....	125, 177	
example script containing.....	125, 177	
TR		
video location top right.....	181	
True/False, False		
under the Task menu item.....	54	
True/False, True		
under the Task menu item.....	53	
Turkish		
keyboard layout.....	370	
Tutorials		
creating.....	129	
importing.....	131	
including information in script.....	133	
naming.....	133, 168	
example script containing.....	169	
popping up.....	130	
U		
Ukrainian		
keyboard layout.....	371	
UNCHECKED		
with a button.....	196	
UNDERLINE.....	103, 153	
UNGLUE.....	195	
Unicode		
and Author.....	31	
Use [English/W.Europe] Keyboard option.....	83	
Use D Window option.....	81	
V		
Variables.....	278	
example script containing.....	278	
example script using variables as counters.....	282	
global.....	283	
Video		
creating clips	237	
display options.....	180	
inserting into lessons	242	
multimedia option.....	247	
options.....	246	
options and commands.....	248	
preparing files.....	226	
WinCALIS support.....	228	
VIDEO.....	180	
options.....	248	

W		
WCA		
Saving a script	29	
Script	38	
WCFNT.INF		
editing	328	
WHILE		
using with variables	282	
Wild cards	148	
& 87, 149		
* 86, 149		
[] 86, 148		
^ 86, 148		
87, 149		
with braces { }	87	
answer sequencing	91	
as literal characters	94, 150	
Definition	85	
example script containing	89	
WinCALIS		
strengths	1	
WinCALIS Author		
Editing Windows		
Directions window	65	
elements of	32, 34	
Interactive (Question) window	66	
Lesson Description window	34	
Popup Directions window	34	
Presentation Text window	35	
Menu		
Edit	44	
File	38	
Format	50	
Help	64	
Language	48	
Task	54	
Tools	55	
Menu Options		
Script	77	
Script----generation options	83	
Overview	30	
WinCALIS Author editor	31	
WinCALIS Commands	150	
Basic Presentation Commands	151	
text attributes	151	
categories of	151	
conditions for general use	151	
Screen Design Commands	169	
Window Manipulation Commands	164	
<POPUP>	165	
<SIZE>	166	
WinCALIS layout		
Menu		
Help	17	
Language	13	
Tutorials	13	
Menu items		
Lesson	11	
Options	12	
opening screen	10	
description of	10	
Opening Screen		
Change directory	9	
Change directory command	30	
Help Menu	9	
Index	9	
No Conversion	9	
Options menu	9, 30	
Text Conversion	9	
Text-CALIS conversion	9	
Options menu		
Window defaults	9	
Window Commands		
available in WinCALIS	110, 166, 177	
choosing TOPLINE	125	
fontsize	106	
Importing Tutorial Files	131	
setting colors	102	
Directions and Tutorial Windows		
example script containing	164	
example script demonstrating rewriting of text	114	
explained	113	
Feedback Diagnostic Window	162	
example script containing	163	
Interactive (Question) Window	161	
example script containing	161	
Overview	159	
Popup Directions and Tutorial Windows	163	
Presentation (Text) Window	162	
example script containing	162	
Window Menu	61	
Windows		
in WinCALIS Author	34	
Y		
Yiddish		
keyboard layout	373	
Z		
Zhuyin Fuhao (Chinese)	298	
Zoom Image		
multimedia option	247	