User's Guide

DCL CODE GENERATOR FOR AutoCAD™ R12

Copyright © 1994, Carlos Legarreta

© Copyright 1994,

Carlos Legarreta
14 Beacon St. Apt. #14
Lawrence, MA 01843
e-mail: legarret@ecs.umass.edu

AutoCAD, DCL, Programmable Dialogue Box Facility (PDB), AutoLISP and ADS are registered trademarks of Autodesk, Inc. *WCEdit* is a registered trademark of Walter Craig and ELSA America, Inc.

TABLE OF CONTENTS

INTRODUCTION ***4***
Features 4
Using this Guide 4
INSTALLATION ***5***
Distribution Files 5
Installing DCG 5
Loading and Executing DCG 5
THE DCL CODE GENERATOR ***6***
1. Status Area 6
2. Control Commands 6
3. Editing Commands 9
4. Predefined and Selected Elements 10
WORKSHOP ***11***
Procedure 11
APPENDIX A Predefined Elements / Attributes ***15***
APPENDIX B Error Messages ***16***
APPENDIX C Revision History ***18***
APPENDIX D AutoLISP Source Code ***19***
APPENDIX E DCL Source Code ***20***

INTRODUCTION

AutoCAD's new Dialogue Control Language (DCL) allows the design and implementation of dialogue boxes in AutoLISP and ADS applications. Programming DCL dialogue boxes for AutoCAD is a time consuming process involving many stages such as design, coding, previewing, and debugging.

The **DCL CODE GENERATOR** (DCG) is an AutoLISP application designed to assist programmers in the development of DCL dialogue boxes. Whether a beginner or experienced programmer, DCG offers an easy-to-learn integrated development environment that makes DCL programming a simple and effortless task. With DCG you don't have to move back and forth between a text editor and AutoCAD to create, preview, and debug your DCL dialogue boxes.

Features

Some of the features offered by DCG are:

- Easy to use graphical interface
- Automatic generation of properly formatted DCL code
- Preview function, which automatically processes and executes a dialogue box
- Direct integration with AutoCAD's *Programmable Dialogue Box* (PDB) facility.
- Context-sensitive online help facility.

Using this Guide

Throughout this guide, various typefaces are used to denote and distinguish special terms and properties. These typefaces are briefly summarized in the following table:

Convention	Description
`Monospace`	Represents text as it appears on-screen or in a program.
`Monospace Bold`	Used for anything that you type (such as **DCG** to start up the DCL CODE GENERATOR)
`[monospace]`	Monospace text enclosed in square brackets indicates cluster type elements from the *Selected Elements* list
`(monospace)`	Monospace text enclosed in parenthesis indicated tile type elements from the *Selected Elements* list
UPERCASE BOLD	Indicates commands, or dialogue names in DCG
Bold	Reperesents an option or indicator in DCG
Italic	Indicates a reference name in DCG (e.g., *Selected Elements*, *DCL Code Settings*, *DCL Work File*), or a special keyword

INSTALLATION

Distribution Files

The complete DCG package consist of the following files:

File	Description
DCG.LSP	AutoLISP program code
DCG.DCL	DCG's DCL dialogue box definitions
DCG.HLP	Help file
DCG.HDX	Help file index
DCG.CFG	Default configuration file
DCG.SLD	Icon file
DCG.PS	Documentation (PostScript® format)
SAMPLE.DCG	Sample dialogue box in .DCG format

All files, with the exception of DCG.PS and SAMPLE.DCG, are required for proper execution of DCG.

Installing DCG

There are two ways DCG can be installed on a system. The first and most simple is by copying the distribution files to one of AutoCAD's directories (e.g., \ACAD12 or \ACAD12\SUPPORT). Make sure the directory you choose is included in the AutoCAD search path environment variable.
(e.g., SET ACAD=C:\ACAD12;C:\ACAD12\SUPPORT; ...).

The second, is by creating a directory for DCG, and copying the distribution files to it. Since DCG needs to find its files during execution, this new directory must be included in the AutoCAD search path environment variable.
(e.g., SET ACAD=C:\ACAD12;C:\ACAD12\SUPPORT; ... ;C:\DCG)

Loading and Executing DCG

To load DCG, go into AutoCAD, and from the "Command:" prompt type,

```
Command: (load "DCG")      <RETURN>
```

Once loaded, the command to start DCG is,

```
Command: DCG               <RETURN>
```

THE DCL CODE GENERATOR

The main DCG dialogue box is divided into four areas; [1] the status area, [2] the control commands area, [3] the editing commands area, and [4] the *Predefined* and *Selected Elements* lists.

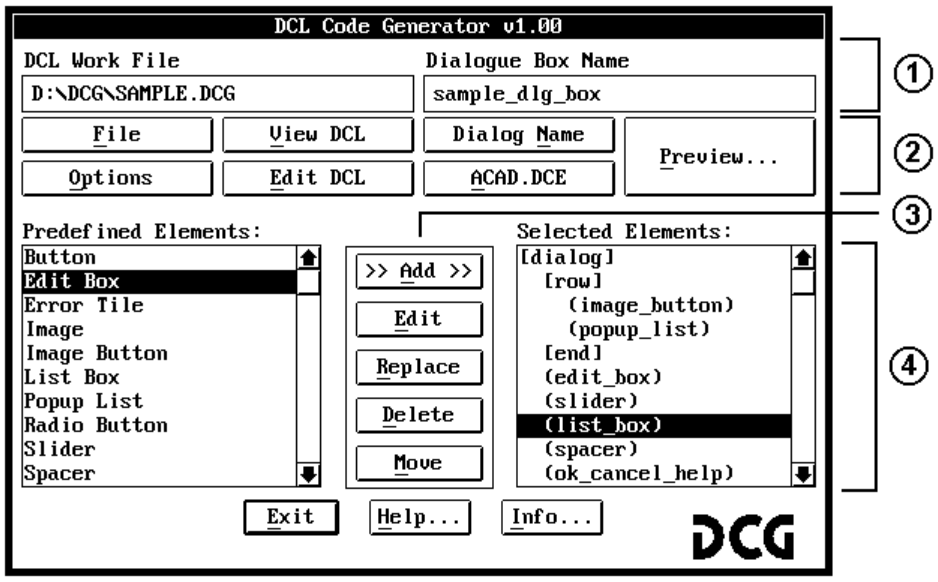


Figure 1 DCG's Main Dialogue Box

1. Status Area

The status area is located at the upper half of the main dialogue box. This area is composed of two text lines which display the current DCL file and dialogue box name. The information shown here is updated automatically to reflect the current settings.

2. Control Commands

The control commands area consists of two rows of buttons which are mainly used to process the current dialogue box and control the DCG editor. The commands available in this area are listed and described below.

File Displays a file menu consisting of the following functions:

Function	Description
NEW	Removes the current dialogue box definition to start a new editing session
OPEN	Opens and loads an existing `.DCG` script file
SAVE	Saves the current dialogue box in both `.DCL` and `.DCG` format
SAVE AS	Same as the **SAVE** function, but prompts first for a file name
MERGE	Appends the current dialogue box code to an existing `.DCL` file
PRINT	Prints the current dialogue box DCL code

Options

Brings up the **OPTIONS** menu, which allows you to define various environment settings for both the DCG editor and the generated DCL code.

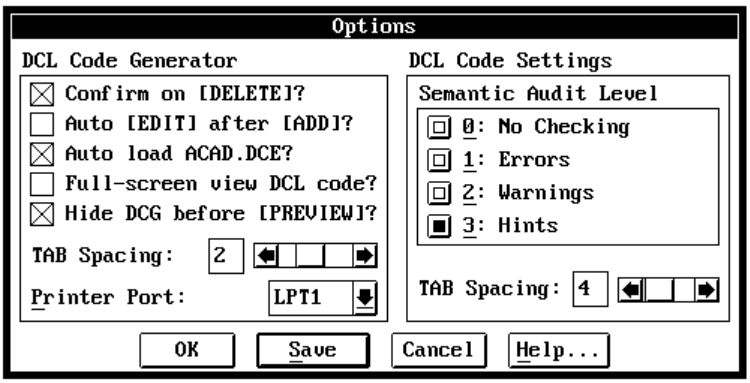


Figure 2 **OPTIONS** Dialogue Box

DCL CODE GENERATOR

Option	Description
Confirm on DELETE?	Prompt before removing a tile or cluster from the *Selected Elements* list
Auto EDIT after ADD?	Automatically **EDIT** a tile after **ADD**ing it to the *Selected Elements* list
Auto load ACAD.DCE?	Automatically load the `ACAD.DCE` error file
Full-screen view DCL code?	View generated DCL code in a full-screen text window instead of a graphical window
Hide DCG before PREVIEW?	Hide DCG before **PREVIEW**ing a dialogue box
TAB Spacing	Number of spaces used to hierarchically indent the DCG pseudo-code in the *Selected Elements* list box
Printer Port	Printer port where your text-capable printer is connected

DCL Code Settings

Option	Description
Semantic Auditing Level	Optional semantic auditing level (0-3) for DCL code. For detailed information on each level, refer to the online help facility
TAB Spacing	Number of spaces used to hierarchically indent the generated DCL code

Save This command, located at the bottom of the **OPTIONS** dialogue box, saves the current options. These options are written to the `DCG.CFG` configuration file, which is automatically loaded every time you start up DCG.

View DCL

Opens a window and displays the actual DCL code of the current dialogue box being edited. You can browse through the code using the arrow keys on the keyboard or the scroll bar located at the right side of the window.

If the **Full-Screen view DCL code?**[1] option is active, the DCL code is displayed in a full-screen text window. This option also offers a **SEARCH** function.

Edit DCL

Allows you to directly edit the DCL code of the current dialogue box using *WCEdit*. Note that any changes made to the code using *WCEdit* will not be reflected on the DCG session, and will be lost. Use this command only with the final version of a dialogue box you have created.

Dialog Name

Allows you to change the name of the current dialogue box. Dialogue box names can only contain letters, numbers, or the underscore character, and must always begin with a letter.

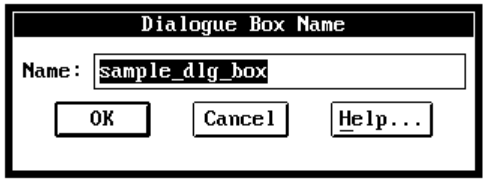


Figure 3 DIALOG NAME Dialogue Box

ACAD.DCE

`ACAD.DCE` is a text file created by the *Programmable Dialogue Box* (PDB) facility when errors occur during the loading process of a DCL file. A list of the errors found on the DCL code along with possible solutions and hints[2], are written to this file. When this file is created, AutoCAD automatically notifies the programmer with an *alert box* and aborts execution of the offending DCL file. By selecting this command, DCG opens a window and displays the contents of this file.

If the **Auto load ACAD.DCE?** option is active, DCG will automatically display this file after the PDB facility creates it.

Preview...

Processes the current dialogue box, by loading it through the PDB facility and displaying it. Depending on the current DCL settings, the PDB facility will report any errors encountered during the loading process. If the **Hide DCG before PREVIEW?** option is active, DCG will hide its main dialogue box before displaying the current dialogue box.

---

[1] This feature requires *WCEdit*™

[2] Only available when `dcl_settings` is set to 3

3. Editing Commands

The editing commands area is located at the center-bottom part of the main DCG dialogue box. The commands in this area are used to create, modify and edit a dialogue box.

>> Add >> Inserts a predefined tile or cluster from the *Predefined Elements* list into the *Selected Elements* list. The new element will be positioned before the selected tile or cluster in the *Selected Elements* list.

If the **Auto EDIT after ADD?** option is active, the **EDIT** command will be executed automatically after the new element has been added.

Edit Use this command to edit the attributes of a tile or cluster from the *Selected Elements* list.

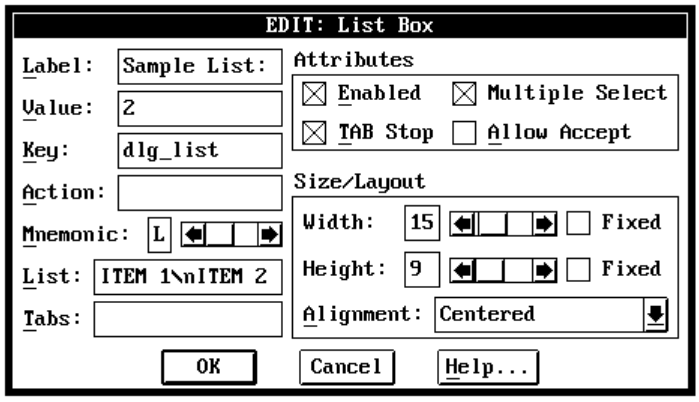


Figure 4 Sample **EDIT** Dialogue Box

Replace Replaces a selected tile from the *Selected Elements* list with one from the *Predefined Elements* list. Note that only tile elements, not clusters, can be replaced.

Delete Removes a tile or cluster from the *Selected Elements* list. If the **Confirm on DELETE?** option is active, DCG will prompt before deleting the element.

Move This command is used to move an tile or cluster in the *Selected Elements* list to another position. When this command is executed, a dialogue box with a copy of the *Selected Elements* list is displayed. From this second list, you can select the new location where the element will positioned.

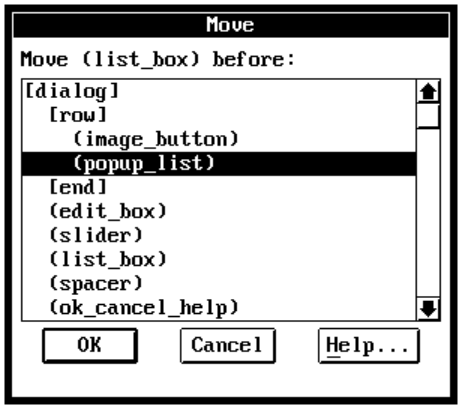


Figure 5 MOVE Dialogue Box

4. Predefined and Selected Elements

The *Predefined Elements* list displays all the available tiles and clusters that can be used to build a dialogue box. Elements shown in initial caps posses predefined attributes which can be modified with the **EDIT** command after they have been **ADD**ed to the *Selected Elements* list. Table 1 in Appendix A shows a list of the predefined elements and their attributes. Elements shown in full caps do not have modifiable attributes.

The *Selected Elements* list displays the elements selected from the *Predefined Elements* list to construct a dialogue box. This list can be though of as the *pseudo-code* of the dialogue box being created.

Elements **ADD**ed to this list will be automatically indented in a hierarchical fashion based in their position within clusters. The default indentation is two spaces, but it can be modified from the **OPTIONS** menu. In this list, cluster type elements are displayed inside square brackets, while tile type elements are enclosed in parenthesis.

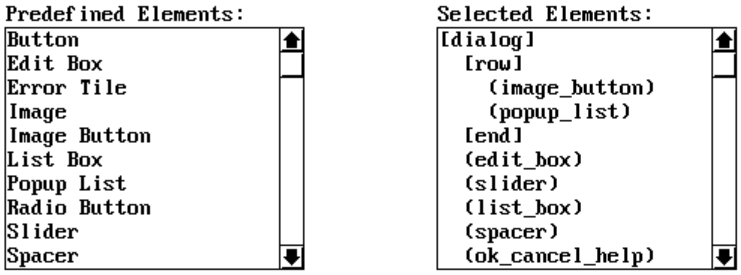
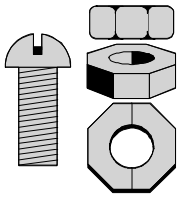


Figure 6 *Predefined* and *Selected Elements* list boxes

WORKSHOP



This workshop is designed to help you familiarize with the DCG editor, and to demonstrate how to construct a sample dialogue box.

Procedure

1. Load and enter the DCL CODE GENERATOR.

Every time DCG is loaded, a standard dialogue box is defined in the *Selected Elements* list. This dialogue box can be considered the most basic definition of a dialogue box in AutoCAD. It only consists of an empty dialogue box with an **OK** button.

- Select the **PREVIEW** button to examine this dialogue box (figure 7).

Figure 7 Standard Dialogue Definition

- Select the **OK** button to return to DCG.

All dialogue boxes will be based on this standard dialogue box definition. With the editing commands, you will be able to construct new dialogue boxes by inserting, editing, and positioning new elements in the *Selected Elements* list.

The dialogue box shown in figure 8, is the sample[3] dialogue box used by AutoCAD's `dlgcolor` command. By following steps 2 through 30, you will be able to construct this dialogue box, and at the same time learn the basic steps to get started with DCG.

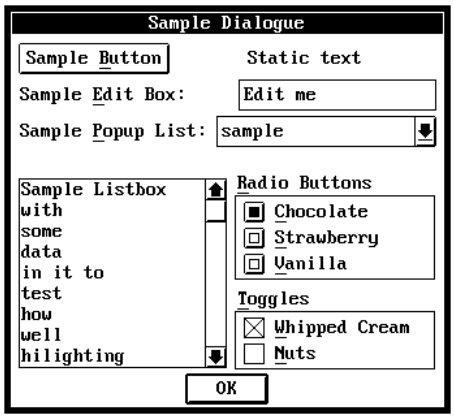


Figure 8 Sample Dialogue Box

[3] A sample `.DCG` script containing the definitions for this dialogue box is included in the DCG distribution package as `SAMPLE.DCG`.

2. Edit the `[dialog]` element by hi-lighting it from the *Predefined Elements* list, and selecting the **EDIT** command. This command will bring up a dialogue box with all the definable attributes for this element.
   a. Enter **`Sample Dialogue`** in the `Label:` field.
   b. Select **OK** to return to the main dialogue box.
3. Add a `Row` cluster element between the `[dialog]` and `(ok_only)` definitions.
   a. Hi-light the `Row` element from the *Predefined Elements* list.
   b. Hi-light the `(ok_only)` element from the *Selected Elements* list.
   c. Select the **ADD** command.
4. **ADD** a `Button` element inside the `[row]` cluster. This time, hi-light the row's `[end]` so that the Button element will be inserted between the `[row]` and its `[end]`.
5. **EDIT** the `(button)` element:
   a. Enter **`sample_button`** in the `Key:` field.
   b. Enter **`Sample Button`** in the `Label:` field.
   c. Enter **`B`** in the `Mnemonic:` field.
   d. Activate the `Fixed` toggle located to the right of the `Width:` field.
6. **ADD** a `Text` element inside the `[row]` cluster and below the `(button)` element.
7. **EDIT** the `(text)` element:
   a. Enter **`Static text`** in the `Label:` field.
8. **ADD** an `Edit Box` element between the `[row]` and the `(ok_button)` elements.
9. **EDIT** the `(edit_box)` element:
   a. Enter **`sample_editbox`** in the `Key:` field.
   b. Enter **`Sample Edit Box:`** in the `Label:` field.
   c. Enter **`E`** in the `Mnemonic:` field.
   d. Set the `Width:` field inside the `Edit...` boxed cluster to **`18`**.
   e. Enter **`Edit me`** in the `Value:` field.
10. **ADD** a `Popup List` element between the `Edit Box` and the `(ok_button)`.
11. **EDIT** the `(popup_list)` element:
    a. Enter **`sample_popup`** in the `Key:` field.
    b. Enter **`Sample Popup List:`** in the Label: field.
    c. Enter **`P`** in the `Mnemonic:` field.
    d. Enter **`This is a\nsample\npopup\nlist\nwith\n several\nitems\nin it`** in the `List:` field.

12. Select the **PREVIEW** command. The resulting dialogue box should look similar to the one shown in figure 9. As you can see, steps 2 through 11 defined the upper half of the final dialogue box.

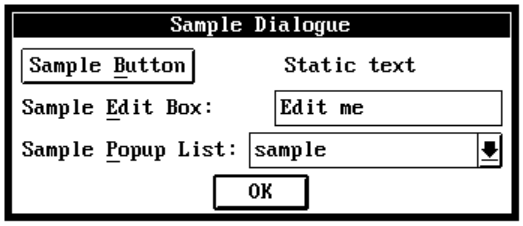


Figure 9 Upper part of the Sample Dialogue Box

13. **ADD** a `SPACER_1` element between the `(popup_list)` and the `(ok_button)`.
14. **ADD** a `Row` cluster element below the `SPACER_1` element.
15. **ADD** a `List Box` element inside the new `[row]` cluster.
16. **EDIT** the `(list_box)` element:
    a. Enter **`sample_listbox`** in the `Key:` field.
    b. Enter **`Sample Listbox\nwith\nsome\ndata\nin it to \ntest\nhow\nwell\hilighting\nworks`** in the `List:` field.
    c. Set the `Width:` field to **`20`**.
    d. Set the `Height:` field to **`7`**.
17. **ADD** a `Column` inside the new `[row]` and below the `(list_box)` element.
18. **EDIT** the `[column]` element:
    a. Set the `Width:` field to **`20`**.
19. **ADD** a `Boxed Radio Column` element inside the `[column]`.
20. **EDIT** the `[br_column]` element:
    a. Enter **`Radio Buttons`** in the `Label:` field.
    b. Enter **`R`** in the `Mnemonic:` field.
21. **ADD** three `Radio Button` elements inside the `[br_column]` cluster.
22. **EDIT** the first `(radio_button)` element:
    a. Enter **`Chocolate`** in the `Label:` field.
    b. Enter **`rad_1`** in the `Key:` field.
    c. Enter **`1`** in the `Value:` field.
    d. Enter **`C`** in the `Mnemonic:` field.
23. **EDIT** the second `(radio_button)` element:
    a. Enter **`Strawberry`** in the `Label:` field.
    b. Enter **`rad_2`** in the `Key:` field.
    c. Enter **`S`** in the `Mnemonic:` field.
24. **EDIT** the third `(radio_button)` element:

a. Enter **`Vanilla`** in the `Label:` field.
b. Enter **`rad_3`** in the `Key:` field.
c. Enter **`V`** in the `Mnemonic:` field.

25. **ADD** a `Spacer` element below the `[br_column]` cluster.

26. **ADD** a `Boxed Column` below the `(spacer)` element.

27. **EDIT** the `[b_column]` element:
    a. Enter **`Toggles`** in the `Label:` field.
    b. Enter **`T`** in the `Mnemonic:` field.

28. **ADD** two Toggle elements inside the `[b_column]`.

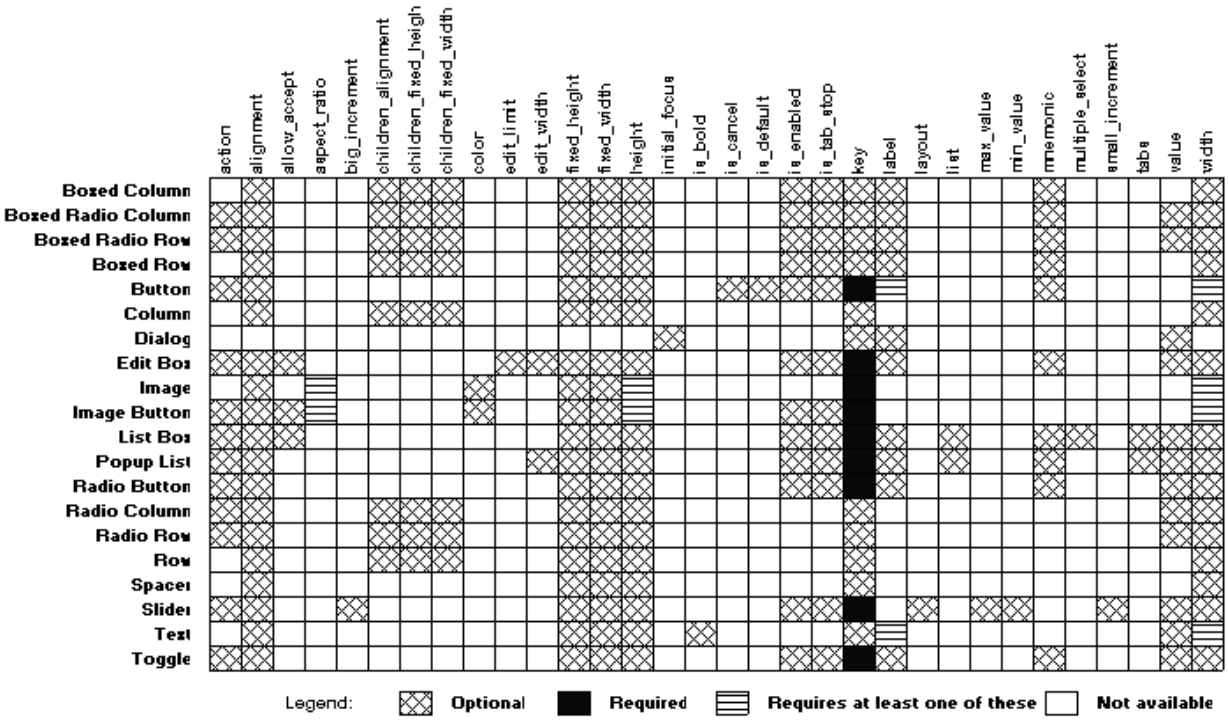
29. **EDIT** the first `(toggle)` element:
    a. Enter **`Whipped Cream`** in the `Label:` field.
    b. Enter **`tog_1`** in the `Key:` field.
    c. Enter **`1`** in the `Value:` field.
    d. Enter **`W`** in the `Mnemonic:` field.

30. **EDIT** the second `(toggle)` element:
    a. Enter **`Nuts`** in the `Label:` field.
    b. Enter **`tog_2`** in the `Key:` field.
    c. Enter **`N`** in the `Mnemonic:` field.

31. **PREVIEW** the dialogue box. Its layout should be similar to the one shown in figure 8.

APPENDIX A Predefined Elements and their Attributes

Table 1



APPENDIX B Error Messages

000: ***Cannot load DCG***
An error occurred during the loading process. This could be caused by corrupt program or configuration files. Solution: Delete all the distribution files, and re-install DCG.

001: ***Environment variable "COMSPEC" not defined***
Some internal functions in DCG require the DOS environment variable `COMSPEC` to be defined. For more information on how to define the `COMSPEC` variable, refer to your DOS manual or your AutoCAD user's guide.

002: ***Cannot save/read TEMP file***
Executing DCG from a full or read-only hard disk could cause this error. DCG will not function properly if it can not create/access its temporary file.

003: ***Invalid dialogue name***
DCL only allows dialogue names to contain letters, numbers, or the underscore character. Also, the names must always begin with a letter.

004: ***File ACAD.DCE not found***
This error occurs when the **ACAD.DCE** command is selected and the actual `ACAD.DCE` file does not exist. This file is automatically created by the *PDB* facility when errors occur during the loading process of a DCL file.

005: ***Cannot preview dialogue box***
There are missing tile attributes in the current dialogue box definition, and for this reason it cannot be previewed.

006: ***Cannot execute PRINT command***
DCG print function is executed through DOS. If DOS returns an error, it will be passed to DCG, and this error message will be displayed.

007: ***This command requires WCEdit™***
WCEdit™ is a programmer's editor for AutoCAD written by Walter Craig, and distributed by ELSA America. Its design permits other applications to access many of its internal features, such as the file viewer, editing functions, etc. If you have a copy of *WCEdit™* in your system, DCG will allow you to preview your generated DCL code through *WCEdit™*'s file viewer. DCG will also let you export the DCL code to *WCEdit™* for further editing.

008: ***Cannot add before [dialog] definition***
Tile and cluster elements can only be inserted between the permanent `[dialog]` definition and its corresponding `[end]`.

009: ***Cannot remove this tile definition***
The permanent `[dialog] definition` and its corresponding `[end] cannot` be removed from the *Selected Elements* list.

010: ***Only tiles can be replaced***

Only tile elements can be replaced by other tile elements. Cluster elements can not be directly replaced.

011: ***Cannot move before* `[dialog]` *definition***
Elements cannot be moved before the permanent `[dialog]` definition.

012: ***Cannot move this tile definition***
This error occurs when the user tries to move the `[dialog]` element, or tries to move a cluster's `[end]` before the actual cluster.

013: ***Could not delete TEMP file***
This error could be caused by file locking, or by not specifying the `COMSPEC` environment variable.

014: ***Cannot edit standard element(s)***
Standard elements, or elements without any modifiable attributes, cannot be edited.

SYS: ***Application Error***
Fatal system or application error. Could be caused by problems in the current environment settings, or in the application itself. Possible solution: Delete the DCG.CFG file, and try to reload the DCG editor.

APPENDIX C Revision History

DCL SCRIPT GENERATOR - DSG v0.90-βeta *February 1, 1994*

- Original pre-release version.
  - Generation of properly formatted DCL code
  - Direct integration with AutoCAD's Programmable Dialogue Box facility
  - On-line help facility
  - Clear and consistent graphical interface

DCL CODE GENERATOR - DCG v1.00 *May 20, 1994*

- Changed name from **DCL SCRIPT GENERATOR** to **DCL CODE GENERATOR**.
- Integrated the **DCL SETTINGS** menu with a new **OPTIONS** menu which allows users to define various environment settings for both the editor and the generated DCL code. The new options are:
  - Confirm on **DELETE**.
  - Automatically **EDIT** a new tile after selecting **ADD.**
  - Automatically load `ACAD.DCE` when created.
  - Select between full-screen[4] or windowed viewing of generated DCL code.
  - Hide DCG's main dialogue box before **PREVIEW**ing a new dialogue.
  - Definable **TAB SPACING** for DCL and DCG hierarchical code indentation.
  - Printer port (`NONE`, `PRN`, `LPT1`, `LPT2`, etc.)
- Replaced **OPEN** and **SAVE** commands with a new **FILE** menu, containing new file-management options such as **NEW**, **OPEN**, **SAVE**, **SAVE AS...**, **MERGE**, and **PRINT**.
- Included new code to link DCG to Walter Craig's *WCEdit* AutoLISP editor. This implementation allows direct editing and/or viewing of the DCL code generated by DCG.
- Double-clicking on one of the *Predefined Elements* will automatically **ADD** that item to the *Selected Elements* list. Double-clicking on one of the *Selected Elements* will automatically execute the **EDIT** command.
- Revised and optimized the source code.
- Included a **SAVE** command on the **OPTIONS** dialogue box, to save the current settings and options to a configuration file, DCG.CFG, which will be loaded automatically by DCG if found in the current directory.
- Improved DCG's error handling. An indexed list of possible errors and their definitions is listed on Appendix B.
- Included a new `Comment` tile which allows direct insertion of comments in the DCL code.

[4] This feature requires Walter Craig's *WCEdit*™

APPENDIX D AutoLISP Source Code

APPENDIX E DCL Source Code