

**USER GUIDE for GRAPHICAL DATABASE  
for CATEGORY THEORY**

**Jeremy Bradbury, Ian Rutherford,  
Matthew Graves and Robert Rosebrugh  
July 1, 2002**

## TABLE OF CONTENTS

### SECTION 1: INTRODUCTION

---

- ▶ About GDCT
- ▶ Development Team
- ▶ Installation
- ▶ Directory Structure

### SECTION 2: CATEGORIES

---

- ▶ Introduction
- ▶ Category File Types
- ▶ CAT File Format
- ▶ CGL File Format
- ▶ Creating Categories
- ▶ Opening Categories
- ▶ Downloading Categories
- ▶ Saving Categories
- ▶ Printing Categories
- ▶ Adding Data
- ▶ Removing Data
- ▶ Changing a Category Name

### SECTION 3: GRAPHICAL DISPLAY OF CATEGORIES

---

- ▶ Introduction
- ▶ Visual Display Controls
- ▶ View GML

### SECTION 4: CATEGORY TOOLS

---

- ▶ Make Confluent
- ▶ Equality of Composites
- ▶ Make Dual Category
- ▶ Initial Object tool
- ▶ Epimorphism tool
- ▶ Coequalizer tool
- ▶ Sum tool
- ▶ Terminal Object tool
- ▶ Monomorphism tool
- ▶ Equalizer tool
- ▶ Product tool

## SECTION 5: FUNCTORS

---

- ▶ Introduction
- ▶ Functor File Types
- ▶ FUN File Format
- ▶ FGL File Format
- ▶ Creating Functors
- ▶ Opening Functors
- ▶ Downloading Functors
- ▶ Saving Functors
- ▶ Printing Functors
- ▶ Functor Animation
- ▶ Diagram Display

## SECTION 6: SETTINGS

---

- ▶ Switching Between Category and Functor Displays
- ▶ Font Settings
- ▶ Category Graphical Settings
- ▶ Functor Graphical Settings
- ▶ Functor Animation Settings
- ▶ Server Settings
- ▶ Endomorphism Limit
- ▶ Displaying the VGJ Controls
- ▶ Displaying the Category Comments
- ▶ Displaying the Functor Comments

## APPENDIX A: MENU ITEM SHORTCUTS

---

- ▶ List of Shortcuts

## APPENDIX B: MISC. INFORMATION

---

- ▶ Version Information
- ▶ Bug Report

# USER GUIDE FOR GDCT

## SECTION 1: INTRODUCTION

---

### About GDCT

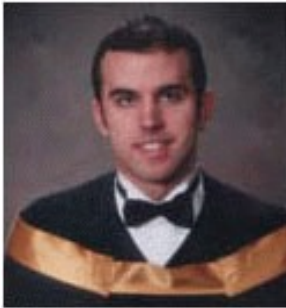
Graphical Database for Category Theory was developed in the summers of 1999, 2000, 2001 and 2002 under NSERC summer research grants. Some of the algorithms used in GDCT were originally developed in A Database of Categories, a C program written by Ryan Gunther and Michael Fleming with supervision by Robert Rosebrugh. This is available at <http://mathcs.mta.ca/research/rosebrugh/dbc/>. The GDCT application is also based on Category Theory Database Tools, a Java applet written by Jeremy Bradbury with supervision by Robert Rosebrugh. This program is available at <http://mathcs.mta.ca/research/rosebrugh/ctdt/>.

This application allows for the creation, editing, and storage of finitely presented categories. Categories can be opened and saved from local files as well as loaded from a specified server. Once a category file is in memory it can be printed or tools for testing properties of objects and arrows can be applied to it. The tools available in this version of GDCT are Make Confluent, Equality of Composites, Make Dual, Initial Object, Sum, Terminal Object, and Product.

This program also allows the display of categories through the use of graph classes that were developed at Auburn University for Visualizing Graphs with Java (VGJ), a tool for graph drawing and graph layout. The original program was heavily modified for displaying categories. It can be found at <http://www.eng.auburn.edu/department/cse/research/graphdrawing/>.

Functors between finitely presented categories can also be created and stored. Functors in memory display their domain and codomain categories, and their action is shown by an animated display.

## Development Team



### **Jeremy Bradbury**

B.Sc. Honours in Computer Science & Mathematics, Mount Allison University

e-mail address: [bradbury@cs.queensu.ca](mailto:bradbury@cs.queensu.ca)

webpage address: <http://www.cs.queensu.ca/home/bradbury/>



### **Ian Rutherford**

B.Sc. Honours in Computer Science & Mathematics, Mount Allison University

e-mail address: [ijrthrf@mta.ca](mailto:ijrthrf@mta.ca)

webpage address: <http://cs.mta.ca/~ijrthrf/>



### **Dr. Robert Rosebrugh**

Professor, Dept. Mathematics and Computer Science, Mount Allison University

e-mail address: [rosebrugh@mta.ca](mailto:rosebrugh@mta.ca)

webpage address: <http://www.mta.ca/~rrosebrugh/>

## Installation

The most current version of Graphical Database for Category is available for download at <http://mathcs.mta.ca/research/rosebrugh/gdct/downloadv20.htm>. There are three installation options. The first option is available only to Windows 95/98/NT/2000 users. Options two and three are available to users of all platforms.

### **Option One: Download Compiled Java Source Files and Java Runtime Environment [Windows Only]**

This includes compiled source files into a jar file. This setup includes a Java Runtime Environment for Windows 95/98/NT/2000 only. This version of the GDCT Setup is \*\*\* MB in size.

### **Option Two: Download Compiled Java Source Files**

The option includes the \*.jar file and sample files, but does not include the JRE. This option will most likely be used by someone who wants to install GDCT under a Linux or Macintosh environment.

Linux Instructions:

You will need the latest JRE from <http://java.sun.com>. From the command prompt type:  
"java -jar gdct.jar."

Macintosh Instructions:

You will need to get the MRJ SDK from <http://developer.apple.com/java/download.html>. Then run the program JBindery. Enter GDCT for the class name. Under the Class Path tab click "Add \*.zip" and then select gdct.jar. Then you can save as to create a double clickable file that is equivalent to a Windows \*.bat file.

### **Option Three: Download Java Source Files and Java Developer's Kit**

The complete Java source files are available for download and may be modified under the terms of the GNU General Public License, Version 2. This download also includes all necessary help files as well as sample category and functor files. It is available as a \*.zip file or as a \*.tar.gz file. In order to compile and interpret Java source files, the Java Developers Kit must be downloaded. JDKs are used by an operating system to compile Java source files and interpret Java class files so that they can be run on the local machine.

## Directory Structure

Upon installation, the Graphical Database for Category Theory, unless changed by the user, will be installed in a directory called GDCT. Inside this directory are the subdirectories `cat`, `cgl`, `fgl`, `fun`, `Help`, `Images`, and `Jre1.1`. The following files are also located inside the main directory :

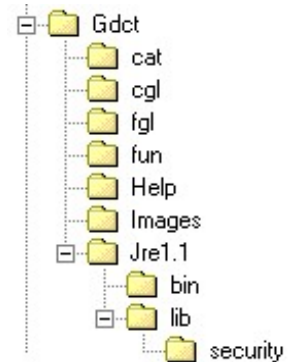
- `GDCT.bat` This is the batch file that the user selects to interpret the java archive file (jar).
- `GDCT.jar` This is the java archive file that is interpreted by the Java Runtime Environment. This file contains all of the compiled Java source files.
- `GDCT.ico` This is an icon file that is used on any shortcuts to the GDCT application.
- `GDCT.ini` This initialization file contains information regarding internal settings, category graphical settings, functor graphical settings, functor animation settings, server settings, as well as a list of the most recently opened files.

The subdirectories `cat` and `cgl` contain sample category files. The subdirectories `fun` and `fgl` contain sample functor files.

The subdirectory `Help` contains all of the help files while the subdirectory `Images` contains all of the images used by the Java application.

The `Jre1.1` subdirectory contains the Java Runtime Environment Version 1.1 which is called by `GDCT.bat` to interpret `GDCT.jar`.

**WARNING:** The user should not modify any files in the subdirectories `Help` or `Images`.



### Introduction

A finitely presented category can be considered a “directed graph with relations between paths of its edges”.

Our notation for category data is:

Objects:

A, B, C, D, . . .

Maps/Arrows:

$e:A \rightarrow A$ ,  $f:A \rightarrow B$ ,  $g:B \rightarrow C$ ,  $h:A \rightarrow C$ . . .

Identity Maps/Arrows:

$1:A \rightarrow A$ , ... (one for each object)

Composition of Maps/Arrows defined by Relations:

$e = 1$ ,  $g*f = h$ , ... (path of arrows)

NOTE: The identity Maps/Arrows are not explicitly shown for every object but they do exist and should not be added manually.

### File Types, Categories

In GDCT, there are three different file types associated with categories:

**\*.cat**

These files are a text only representation of a category. These files can be both opened and saved. Since a graphical representation of the category is not saved one is generated randomly upon opening the \*.cat file.

**\*.cgl**

These files contain both text and graphical representations of a category and can be both opened and saved. They are a concatenation of a .cat file and a .gml file.

**\*.gml**

These files contain only a graphical representation of a category. These files can only be saved (and may be used by the VGJ applet), they can not be opened.

### CAT File Format

The text representation of categories is stored in the following format. Any line that is a comment must start with the "#" character. On the line following the word "category" is the category name. Objects are listed between the words "objects" and "arrows". Arrows are listed between the words "arrows" and "relations". Relations are listed after the word "relations".

NOTE: The words "category", "objects", "arrows", "relations", and "gml" are key words and cannot be used as the name of the category or as the name of an object or arrow within the category.

Objects and arrows can be a string of any length and can contain any characters excluding ":", "-", ">", "\*", "#",



":", "=", ",", " ".

Relations are composed using the composition symbol "\*".

An example of a typical CAT file can be found in Appendix A.

## CGL File Format

A CGL file contains the information for both a text representation and a graphical representation.

The text representation is the same as the file format for CAT files. Any line that is a comment must start with the "#" character. On the line following the word "category" is the category name. Objects are listed between the words "objects" and "arrows". Arrows are listed between the words "arrows" and "relations". Relations are listed after the word "relations".

*NOTE:* The words "category", "objects", "arrows", "relations", and "gml" are key words and cannot be used as the name of the category or as the name of an object or arrow within the category.

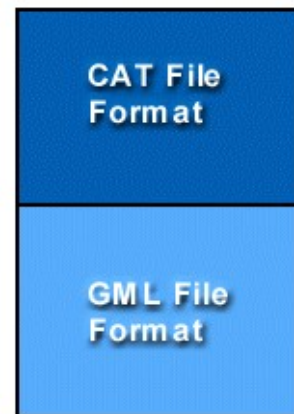
Objects and arrows can be a string of any length and can contain any ASCII characters excluding ":", "-", ">", "\*", "#", ".", "=", ",", " ".

Relations are composed using the composition symbol "\*".

The graphical representation uses a format known as gml which was developed at Auburn University for use with the Visualizing Graphs with Java (VGJ) project. It is incorporated into the cgl format by placing it at the end of the file after the keyword "gml". For more information consult the VGJ Documentation.

An example of a typical CGL file can be found in Appendix A.

## CGL File Format



## Creating Categories

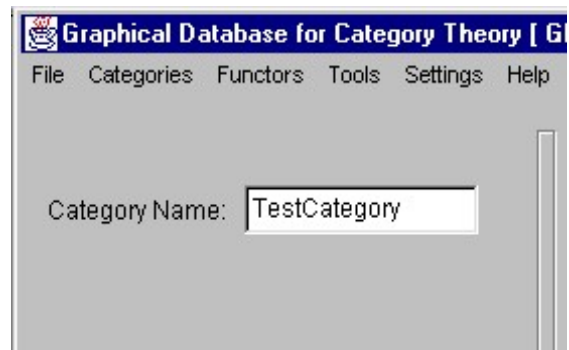
To create a category in memory the user selects Categories | Create Category.

The names of objects and arrows can be strings of any length except for reserved words and may contain any ASCII characters excluding ":", "-", ">", "\*", "#", ".", "=", ",", " ".

Enter a (descriptive!) name in the "Category name" box and hit Enter. Type the names of objects in the "Object Name" box which now appears, completing each with Enter. Press the "Objects Complete" button when done.

During creation of the category the objects and arrows are displayed in the Category Display Window. Objects may be moved about by mouse action at any time. Click on an object and drag it to a new position.

*WARNING:* No two objects may have the same name. If a duplicate object name is entered, an error message will pop up and the new object will not be included in the category.

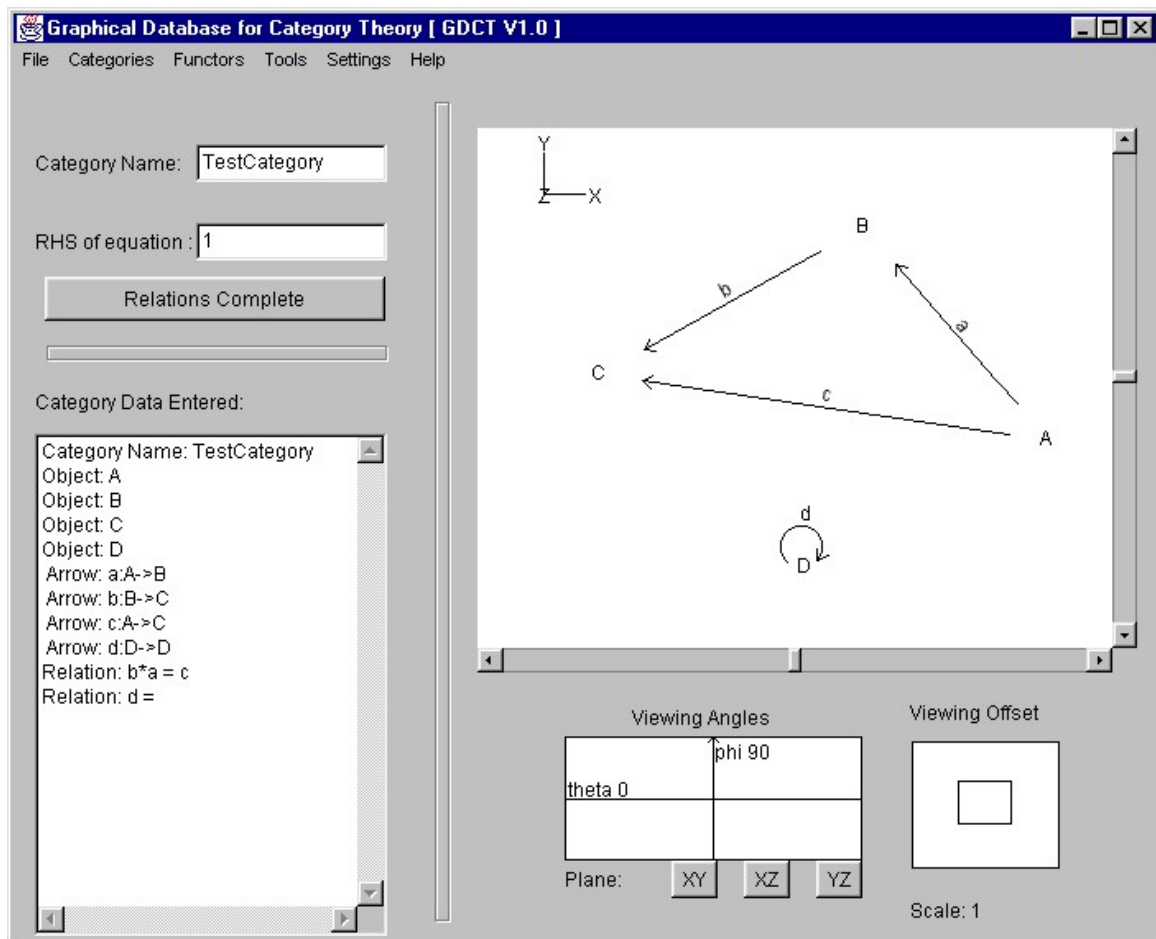


Entering arrows is very similar to entering objects. Type the names of arrows in the "Arrow Name" box which now appears, completing each with **Enter**. After the name of an arrow is entered, the user is prompted for the domain and then the codomain of the arrow. Press the "Arrows Complete" button when done.

As arrows are entered they are displayed in the Category Display window.

**WARNINGS:** No two arrows may have the same name. The domain and codomain of each arrow must be a declared object. An arrow cannot be named "1" since this symbol is reserved by the program for the identity arrow.

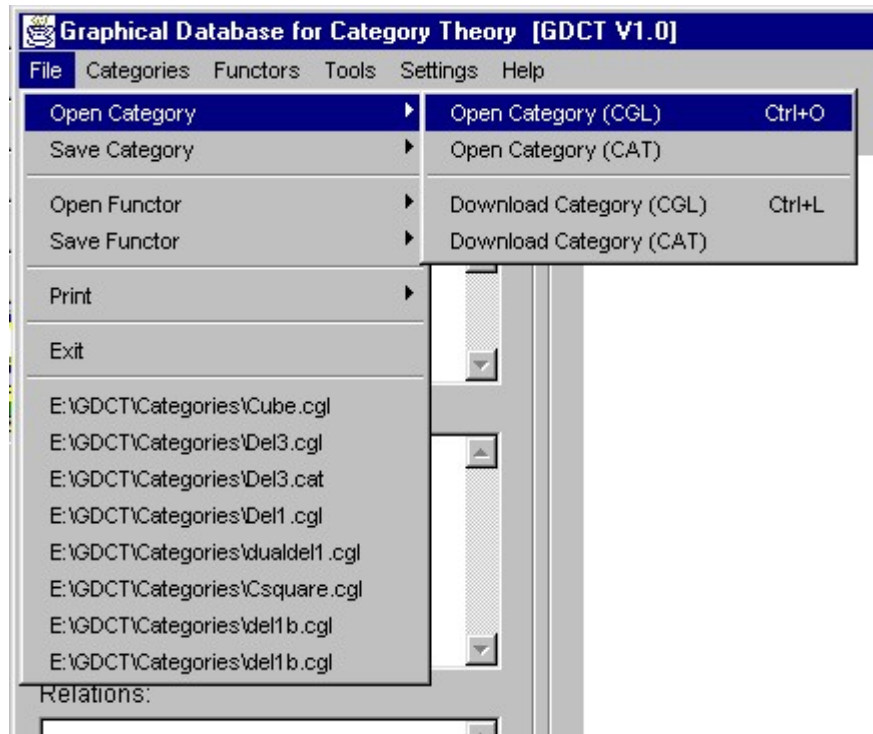
Next the user enters the equations among composites of the generating arrows. The "LHS of equation" box appears and the user must enter a composable path of arrows (separated by the composition symbol "\*"). In the "RHS of equation" box another composable path is entered. After the equations have been entered press the "Equations complete" button.



**WARNING:** Only valid equations are accepted. An equation is invalid if it contains arrows that are not in the category, or an undefined composition, or if the domain and codomain of the left side do not equal the domain and codomain of the right side.

## Opening Categories

In order to open a text category file (.CAT) select File | Open Category | Open Category(CAT) or Categories | Open Category | Open Category(CAT). This opens a file selection window. Opening graphical category files (.CGL) is similar. An alternative method to open a .cgl category file is to press CTRL+O. To open a category simply double click on the category's name or click once on the category name and press the OK button.



If there is any error in the format of the selected file an error message is displayed and the open is aborted.

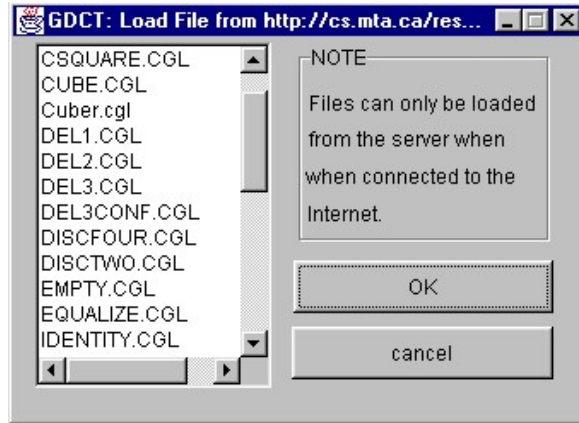
The most recent category files can also be opened by clicking on one of their names in the File menu.

## Downloading Categories

In order to download a text category file (.cat) select File | Open Category | Download Category(CAT) or Categories | Open Category | Download Category(CAT).

Downloading a graphical category file (.cgl) is similar, select File | Open Category | Download Category(CGL) or Categories | Open Category | Download Category(CGL). An alternative method to download graphical category files is to press CTRL+D.

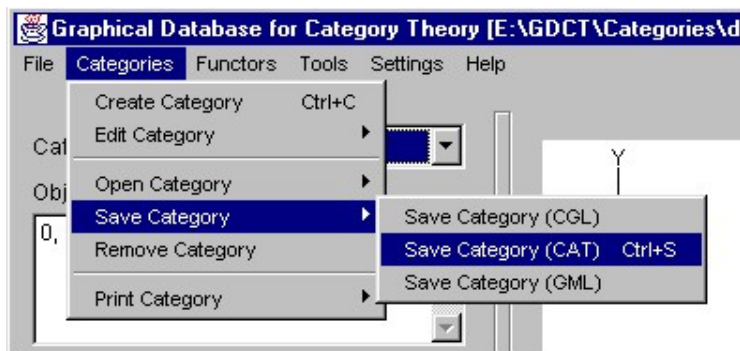
Choosing any of the above options will display a list of categories in a selection box in a new frame. The categories listed are located on the server specified in the frame's title bar. To select a category to download simply double click on the category's name or click once on the category name and press the OK button. Either of these methods will display the downloaded category. If the program cannot locate the files on the server an error message will be displayed. This could result from the user not being connected to the Internet or as a result of a server error.



To change the settings for downloading categories, select Settings | Set Server. This displays a dialog for changing the default server and directories from which categories and functors can be loaded using the appropriate load command.

### Saving Categories

In order to save a category file (.cgl) select File | Save Category | Save Category(CGL) or Categories | Save Category | Save Category(CGL). This opens a file selection window. Saving text category files (.cat) or simply the VGJ graph file (.gml) is similar. A shortcut to save a .cgl category file is to press CTRL+S. To save a category to an existing file simply double click on the category file name. A new file is created by typing its name in the dialogue and clicking OK.



### Printing Categories

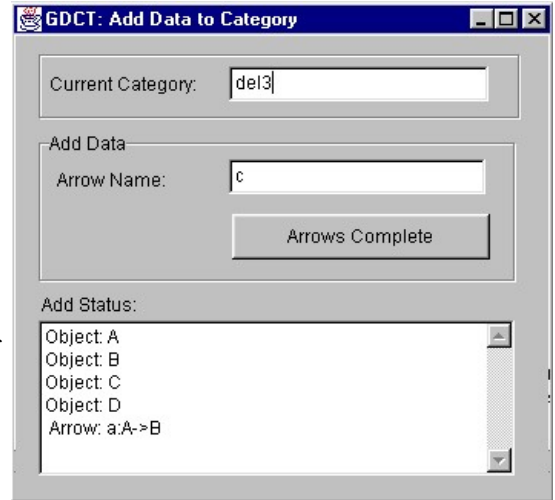
To print the text version of the currently displayed category, select File | Print | Print Category (Text). To print the graphical version select File | Print | Print Category (Graphical).

To save the graphical version of the currently displayed category to a postscript file, select File | Print | Print Category (Postscript).

## Adding Data

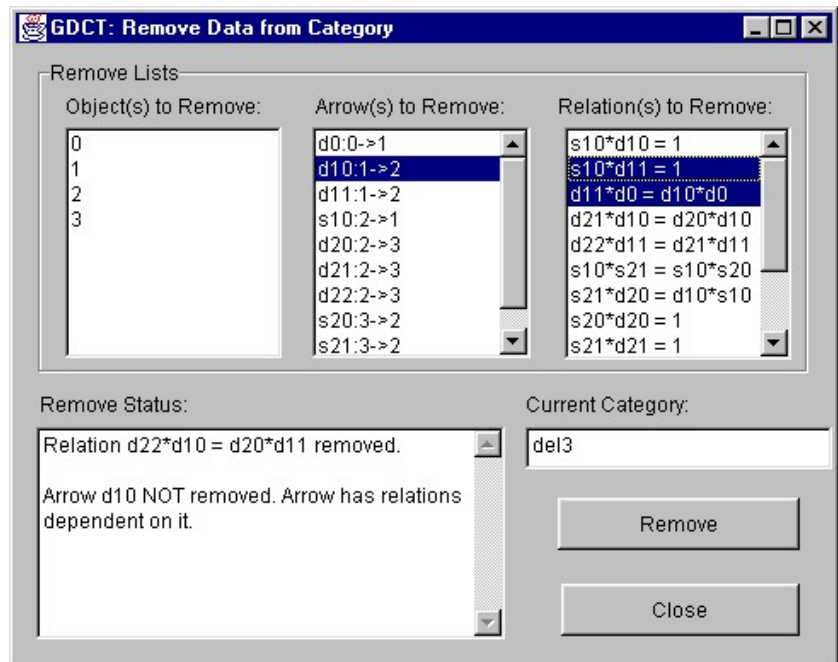
Selecting Categories | Edit Category | Add Data opens a dialogue which allows addition of objects, arrows and relations to the current category. To add objects type their names in the box and hit enter. When finished click Objects Complete, then arrows may be added: type the name, then the domain and codomain; when finished click Arrows Complete. To add relations the left and right sides must be specified; when finished click Relations Complete.

**WARNING:** The domain and codomain of a new arrow must already exist, and the sides of a relation must be composable paths. Adding data only affects the version of a category in memory. To save any changes to disk use the Save Category dialogue.



## Removing Data

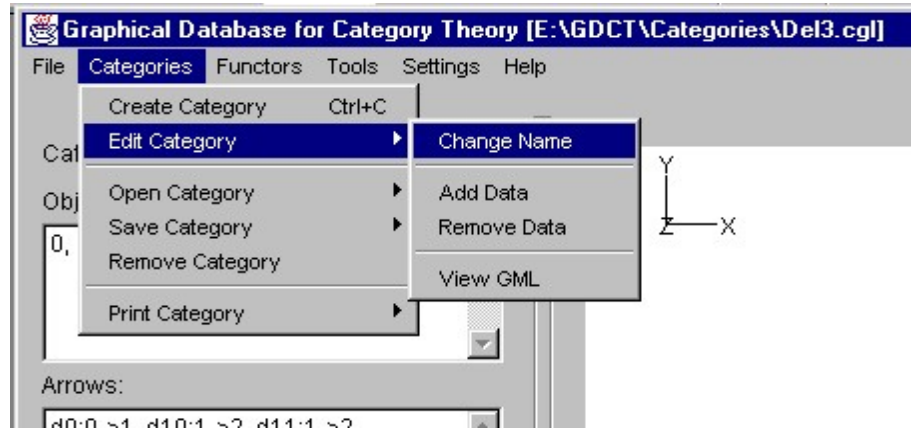
Selecting Categories | Edit Category | Remove data opens a dialogue with lists of the objects, arrows and relations of the current category. Select items from the lists and click Remove to delete the data from the category.



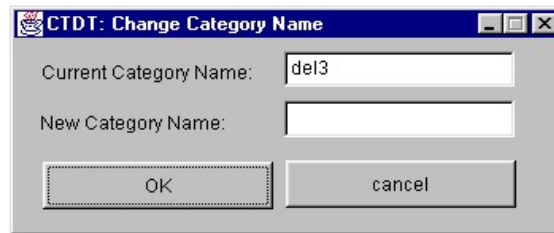
**WARNING:** An arrow can be removed only if it does not appear in any current relations. An object can be removed only if it is neither the domain nor codomain of a current arrow. Removing data only affects the version of a category in memory. To save any changes to disk use the Save Category dialogue.

## Changing a Category Name

To change the name of the currently displayed category, the user selects Categories | Edit Category | Change Name.



Selecting this option will display a dialog in which the user can enter the new name.

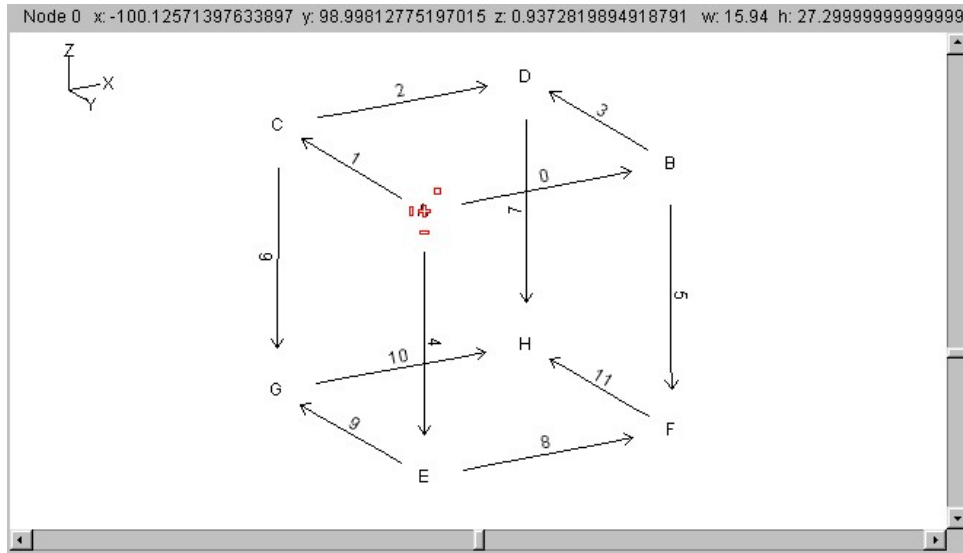


## SECTION 3: GRAPHICAL DISPLAY OF CATEGORIES

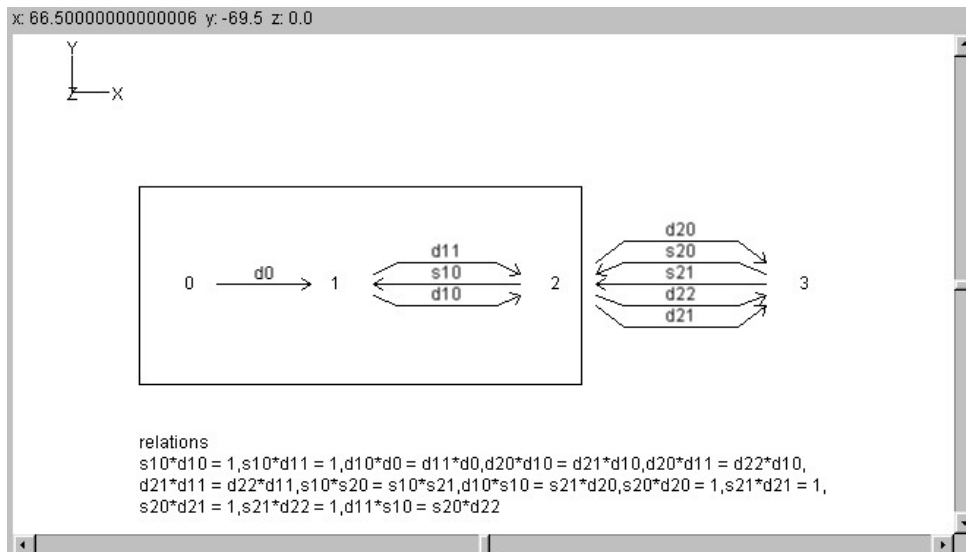
### Introduction

The controls for the graphical display of categories are explained under the title "Visual Display Controls". This section deals with manipulating the graphical display of a category.

There are two ways to select an object or arrow. The first is to click directly on the object or arrow so that it becomes highlighted red.



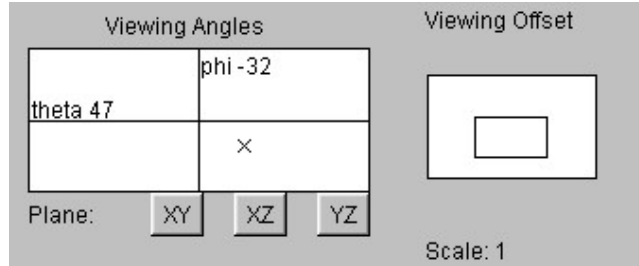
The second way is to click and hold the left mouse button down before dragging a square around the desired object(s) or arrow(s).



Once an object or arrow has been selected, it can be moved by holding down on it and pulling it to another position on the graphical display canvas.

## Visual Display Controls

The controls for the graphical display of categories were developed at Auburn University as part of the a 3D package that is used by GDCT in the display of categories. The visual controls can be hidden from view by selecting Settings | Show VGJ Controls. The visual controls available are:



### *Viewing Angles*

The viewing angle can be changed by using the control below the left side of the graphical display. Click and hold the "X" to rotate the view to any 3D angle. The current orientation of the category displayed can be determined by the XYZ axis displayed in the top left corner of the category display area.

### *Viewing Plane*

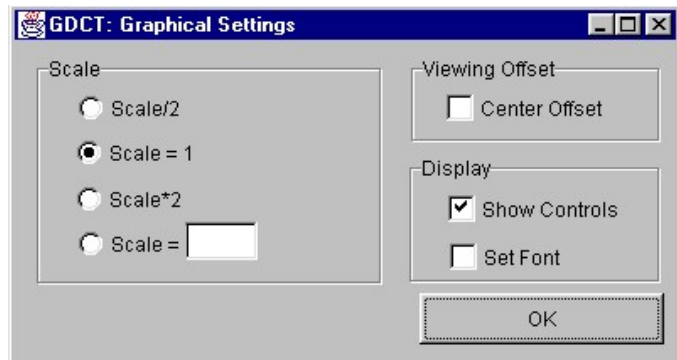
The plane that the category is being viewed in can be changed by pressing one of the three buttons below the "Viewing Angles" control window in the bottom right corner of the main screen. The user can select the XY plane, XZ plane, or the YZ plane.

### *Viewing Offset*

The viewing offset is located in the bottom right hand corner of the main window. It is basically a virtual screen that allows the user to click and hold on the "box" to move the viewable canvas area without using the scroll bars. The viewing offset can be automatically centered by selecting Settings | Category Graphical Settings and clicking the check box by "Center Offset".

### *Scale*

The scale that the category is being viewed at is displayed directly below the "Viewing Offset" controls. The scale can be changed by selecting Settings | Category Graphical Settings. The user can decrease the scale by a factor of two, increase the scale by a factor of two, set the scale to one, or set a user defined scale.



### *Set Font*

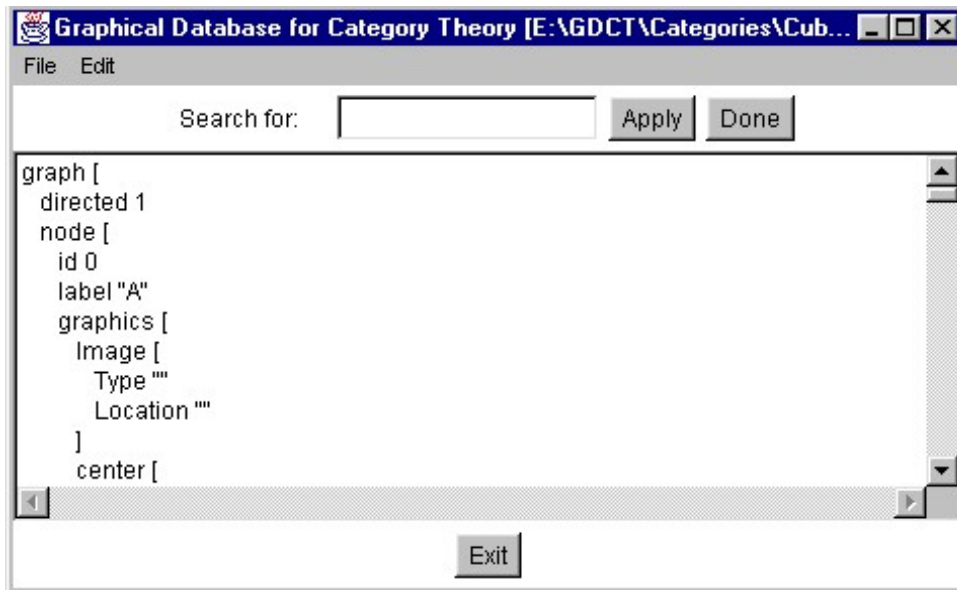
The font size and face can be changed by selecting Settings | Category Graphical Settings and clicking the check box in front of "Set Font". This will display a dialog in which the user can type in the desired face and size for the font. Click "Apply" to save these changes or click "Cancel" to keep the current settings.



## View GML

GML is the text representation of the graphical display of the current category. To view the GML code select Categories | Edit Category | View GML. Choosing this menu option will display a frame containing the GML code.

Each object is stored as a node. The relations are stored as a single node. The arrows are stored as edges between nodes. Multiple arrows between nodes are stored as one edge with commas separating the names of the individual arrows.

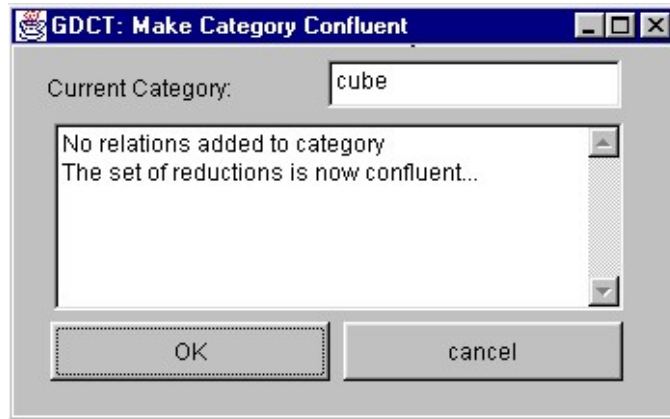


There are several tools to help viewing of the GML:

- Edit | GoToTop
- Edit | GoToBottom
- Edit | GoToLine
- Edit | Search

### Make Confluent

Select Tools | Make Confluent. This command applies the Knuth-Bendix algorithm to the currently displayed category with arrows ordered in their storage order.

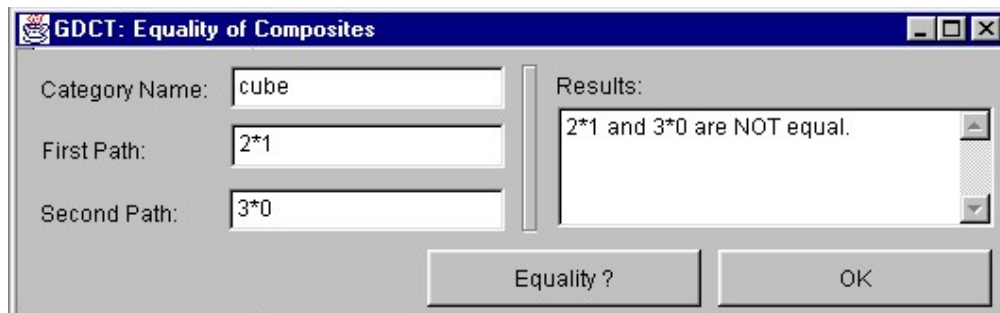


Relations which are added to the current category are reported. To add these relations to the current category click "Ok", or click "Cancel" to keep the category in its current form.

*NOTE:* Most of the other category tools in GDCT require confluent categories in order to function correctly. Make sure to use the "Make Confluent" tool whenever you plan to use other category tools.

### Equality of Composites

To test two (possibly composite) arrows for equality in the current category select Tools | Equality of Composites. A dialogue allows entry of two paths.



*WARNING:* The category must be confluent for valid results. Domains and codomains of paths must match.

### Make Dual Category

This tool will create the opposite category of the currently displayed category. Select Tools | Make Dual. A dialogue window opens prompting for the name of the dual. After the user provides the name the dual category is displayed.

*WARNING:* The opposite category is NOT SAVED automatically when it is created; use File | Save Category if you wish to save the dual.

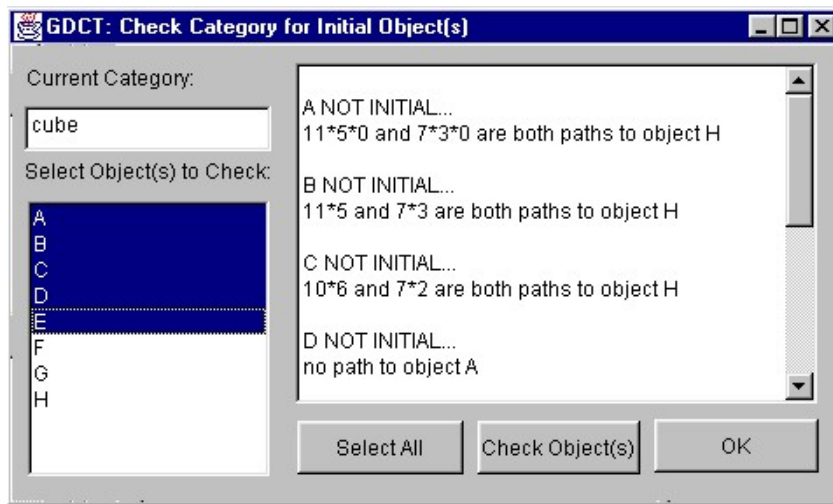
## Isomorphism

Selecting Tools | Isomorphism? opens a dialogue window allowing the input of up to two paths to be checked for isomorphisms. The user can enter one path and have the program search for an inverse, or specify a second path to be checked as an inverse. Alternately, the user can select two objects to check for isomorphism and the tool will search through all paths between them to find inverses.

*WARNING:* The test is only valid on a confluent category. (If in doubt use the Make Confluent tool).

## Initial Object

Select Tools | Initial Object?. This command tests objects of the current category (chosen by a dialogue) for initiality. For each selected object which is not initial a reason is given.



*WARNING:* The test is only valid on a confluent category. (If in doubt use the Make Confluent tool).

## Epimorphism

Selecting Tools | Epimorphism? opens a dialogue window allowing input of a path to check as an epimorphism. Additionally, the "Check all paths" checkbox may be selected to check all paths in the current category.

*WARNING:* The test is only valid on a confluent category. (If in doubt use the Make Confluent tool).

## Coequalizer

Selecting Tools | Coequalizer? opens a dialogue window allowing the specification of first and second paths, and input of a path to be tested as a coequalizer for the two paths. Additionally, the "Check all paths" checkbox may be checked allowing selection of a possible coequalizer object in the current category from a drop-down list. All paths to this object will then be tested as coequalizers.

*WARNING:* The domain of each path and the codomain of each path must be the same. The codomain of the two paths must be the same as the domain of the coequalizer path. The test is only valid on a confluent category. (If in doubt use the Make Confluent tool).

## Sum

Selecting Tools | Sum? opens a dialogue window allowing selection of a possible sum object from the current category from a drop-down list and specification of (possibly composite) first and second injections, then testing the cospan for being a sum.

*WARNING:* The codomain of each injection must be the possible sum object. The test is only valid on a confluent category. (If in doubt use the Make Confluent tool).

## Terminal Object

Select Tools | Terminal Object?. This command tests objects of the current category (chosen by a dialogue) for being terminal. For each selected object which is not terminal a reason is given.

*WARNING:* The test is only valid on a confluent category. (If in doubt use the Make Confluent tool).

## Monomorphism

Selecting Tools | Monomorphism? opens a dialogue window allowing input of a path to check as a monomorphism. Additionally, the "Check all paths" checkbox may be selected to check all paths in the current category.

*WARNING:* The test is only valid on a confluent category. (If in doubt use the Make Confluent tool).

## Equalizer

Selecting Tools | Equalizer? opens a dialogue window allowing the specification of first and second paths, and input of a path to be tested as an equalizer for the two paths. Additionally, the "Check all paths" checkbox may be checked allowing selection of a possible equalizer object in the current category from a drop-down list. All paths from this object will then be tested as equalizers.

*WARNING:* The domain of each path and the codomain of each path must be the same. The domain of the two paths must be the same as the codomain of the equalizer path. The test is only valid on a confluent category. (If in doubt use the Make Confluent tool).

## Product

Selecting Tools | Product? opens a dialogue window allowing selection of a possible product object from the current category from a drop-down and specification of (possibly composite) first and second projections, then testing the span for being a product.

*WARNING:* The domain of each projection must be the possible product object. The test is only valid on a confluent category. (If in doubt use the Make Confluent tool).

## Introduction

A functor is a “structure preserving” interpretation of one finitely presented category into another.

A functor preserves the structure of a category because it sends:

Objects to Objects  
Arrows to Arrows

and it preserves the composition that is provided by relations.

## Functor File Types

In GDCT there are two different file types that are associated with functors.

### **\*.fun**

These files are a text only representation of a functor. These files can be both opened and saved. Since a graphical representation of the two categories involved is not saved, the graphical representations are generated randomly upon opening the \*.FUN file.

### **\*.fgl**

These files contain both a text and graphical representation of the categories involved in a given functor as well as the functor information itself. This type of file can be both opened and saved.

## FUN File Format

The text representation of functors is stored in the following three part format.

Part one consists of a category stored in the CAT file format.

Part two consists of a second category stored in the CAT file format.

Part three consists of the functor information. In this part, any line that is a comment must start with the "#" character. On the line following the word "functor" is the functor name. The mapping of objects to objects is listed between the words "objects" and "arrows". The mapping of arrows to paths is listed after the word "arrows".

*NOTE:* The words "category", "functor", "objects", "arrows", and "gml" are key words and cannot be used as the name of the categories, functors between them or as the name of an object or arrow within categories.

An example of a typical FUN file can be found in Appendix A.

## FGL File Format

The text representation of functors is stored in the following three part format.

Part one consists of a category stored in the CGL file format.

Part two consists of a second category stored in the CGL file format.

Part three consists of the functor information. In this part, any line that is a comment must start with the "#"

character. On the line following the word "functor" is the functor name. The mapping of objects to objects is listed between the words "objects" and "arrows". The mapping of arrows to paths is listed after the word "arrows".

NOTE: The words "category", "functor", "objects", "arrows", and "gml" are key words and cannot be used as the name of the categories, functors between them or as the name of an object or arrow within categories.

An example of a typical FGL file can be found in Appendix A.

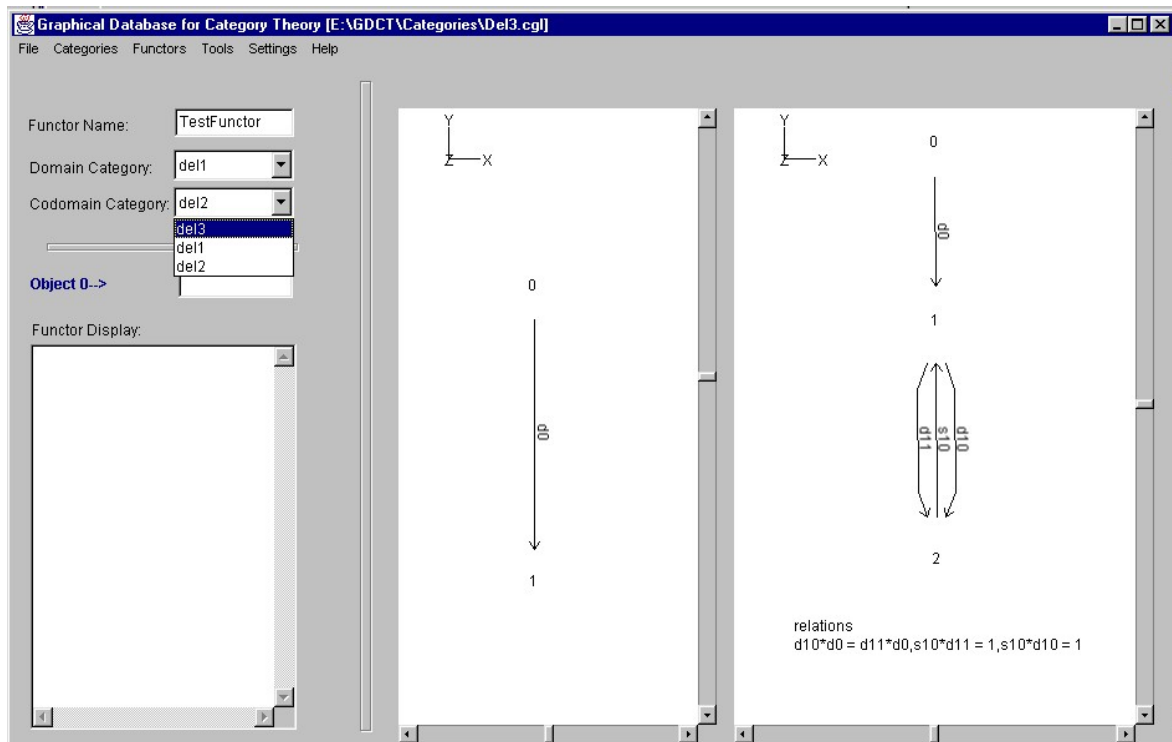
### Creating Functors

To create a Functor in memory the user selects Functors | Create Functor.

First the domain and codomain categories must be selected from those in memory using the drop-down category lists. The categories will be displayed in two windows.

Enter a (descriptive!) name in the "Functor Name" box and type Enter.

### FGL File Format



For each object type the name of its image under the functor in the box which now appears and type Enter. For each arrow type the name of its image under the functor (which may be a composite arrow) in the box and type Enter.

During creation of a functor its action is displayed in the Functor display box.

**WARNING:** The image of an arrow must have domain and codomain which are the images of the domain and codomain of the arrow.

This completes the creation of the new functor. Following creation it is possible to animate the display of the

functor using the buttons below the category display windows.

*WARNING:* The newly created functor is stored in memory. To save the functor to disk select File | Save Functor or Functors | Save Functor. The functor will be saved in the format selected using a save dialog.

## Opening Functors

In order to open a text functor file (.fun) select File | Open Functor | Open Functor(FUN) or Functors | Open Functor | Open Functor(FUN). This opens a file selection window. Opening graphical functor files (.fgl) is similar. A shortcut to open a .fgl functor file is to press CTRL+F. To open a functor simply double click on the functor's name or click once on the functor name and press the OK button.

If there is any error in the format of the selected file an error message is displayed and the open is aborted.

## Downloading Functors

In order to download a text functor file (.fun) select File | Open Functor | Download Functor(FUN) or Functors | Open Functor | Download Functor(FUN).

Downloading a graphical functor file (.fgl) is similar, select File | Open Functor | Download Functor(FGL) or Functors | Open Functor | Download Functor(FGL).

Choosing any of the above options will display a list of functors in a selection box in a new frame. The functors listed are located on the server specified in the frame's title bar. To select a functor to download simply double click on the functor's name or click once on the functor name and press the OK button. Either of these methods will display the downloaded functor. If the program cannot locate the files on the server an error message will be displayed. This could result from the user not being connected to the Internet or as a result of a server error.

To change the settings for downloading functors, select Settings | Set Server. This displays a dialog for changing the default server and directories from which categories and functors can be loaded using the appropriate load command.

## Saving Functors

In order to save a functor in memory as a graphical file (.fgl) select File | Save Functor | Save Functor(FGL) or Functors | Save Functor | Save Functor(FGL). This opens a file selection window. To save a functor to an existing file simply double click on the functor file name. A new file is created by typing its name in the dialogue and clicking OK.

Saving functor files as text only (.fun) is similar.

## Printing Functors

To print the text version of the currently displayed functor, select File | Print | Print Functor (Text).

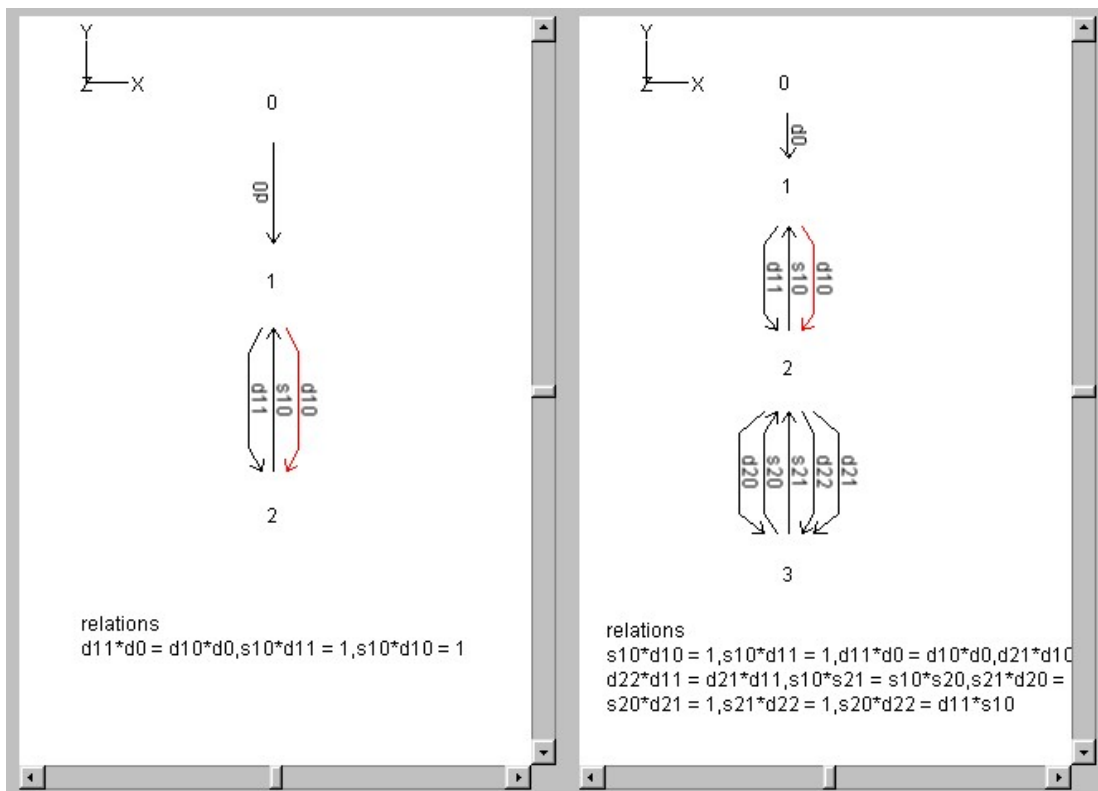
## Functor Animation

The mappings on objects and arrows defined by a functor can be highlighted through the use of animation. There are two ways to animate a functor.



The first way is to use the **Back** and **Forward** buttons below the graphical display of the functor to step through all of the mappings. The animation can be restarted by pressing the **Reset** button.

The second way to animate a functor is to press the **Auto Animate** button which will animate all of the mappings at the animation speed indicated in **Settings | Functor Animation Settings**.



## Diagram Display

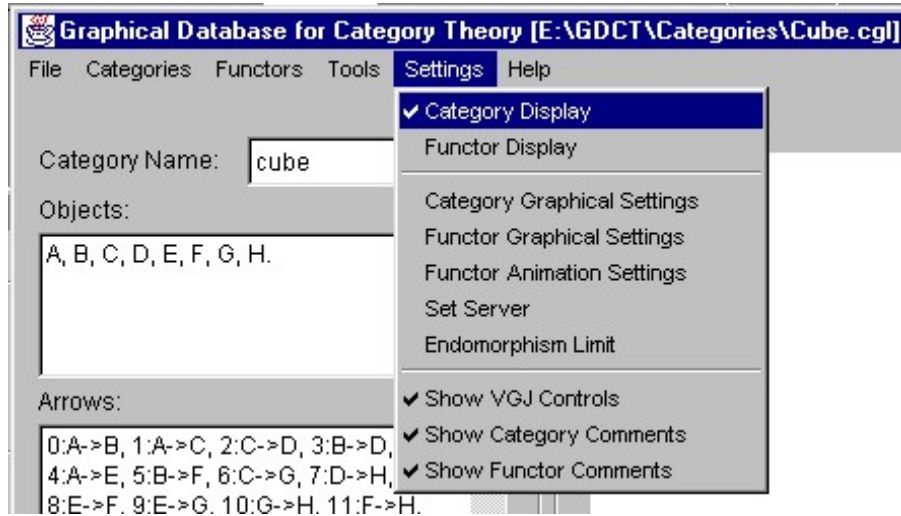
Functors can also be viewed as diagrams, with the graphical display taken from the domain category and the object and arrow names taken from the codomain category. To view a functor as a diagram, click the “Diagram Display” button in the top left corner when a functor is open. To switch back to normal functor display mode, click the “Two Category Display” button.

*NOTE:* When first opened, a functor’s diagram display has a newly created graph with object positions taken from the domain category. A diagram graph is NOT saved in either functor file format.



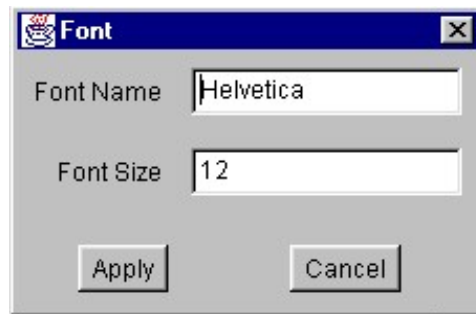
Switching Between Category and Functor Displays

Selecting either of Settings | Category Display or Settings | Functor Display allows the user to switch between the display of categories and the display of functors.



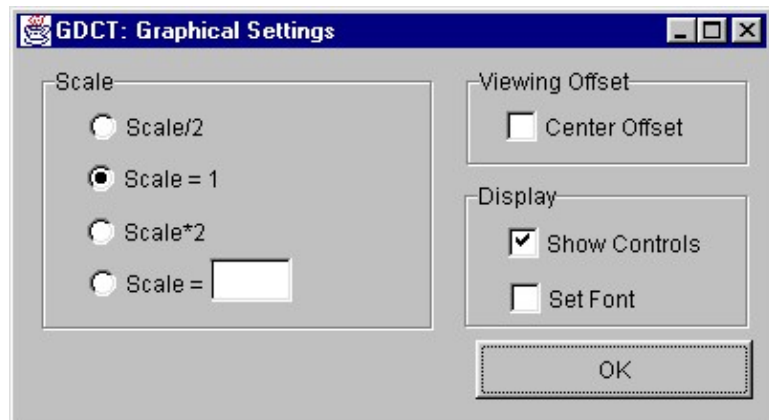
Font Settings

Selecting Settings | Category Graphical Settings and then checking the box in front of “Set Font”. This displays a dialog in which the user can change the font settings within the program.



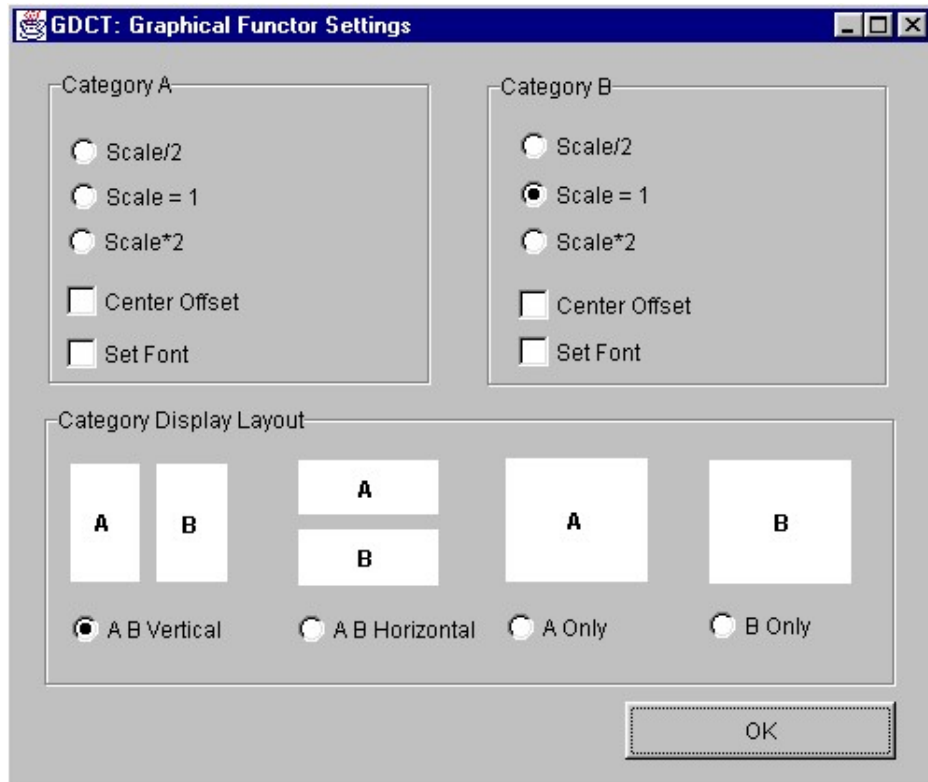
Category Graphical Settings

Selecting Settings | Category Graphical Settings displays a dialog for changing the graphical settings in the category display.



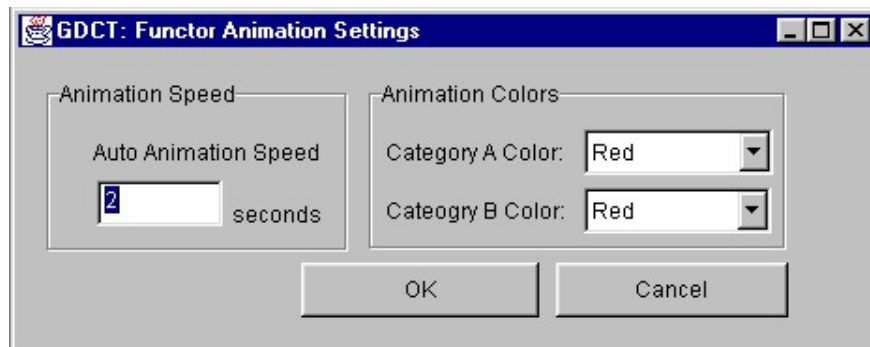
## Functor Graphical Settings

Selecting Settings | Functor Graphical Settings displays a dialog for changing the graphical settings in the functor display.



## Functor Animation Settings

Selecting Settings | Functor Animation Settings displays a dialog for changing the animation delay time (default 2 seconds) and the animation colors for the domain and codomain categories.

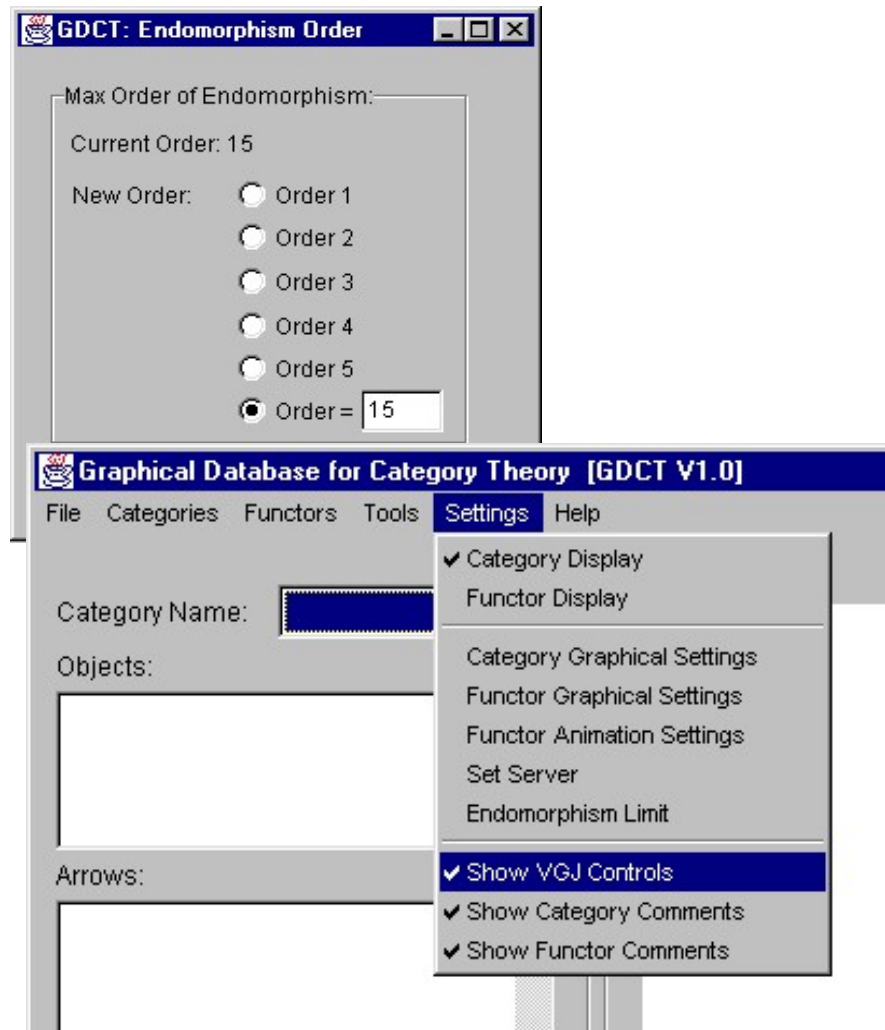


## Server Settings

Selecting Settings | Set Server displays a dialog for changing the default server and directories from which categories and functors can be loaded using the appropriate load command.

## Endomorphism Limit

Selecting Settings | Endomorphism Limit displays a dialog for changing the endomorphism limit.



## Displaying the VGJ Controls

If Settings | Show VGJ Controls is selected it displays the VGJ controls in the category display. If deselected, it hides them.

## Displaying the Category Comments

If Settings | Show Category Comments is selected it displays the comments in the category display. If deselected, it hides them.

## Displaying the Functor Comments

If Settings | Show Functor Comments is selected it displays the comments in the functor display. If deselected, it hides them.

## APPENDICES

### APPENDIX A: MENU ITEM SHORTCUTS

---

The following short cuts to menu items are supported in Graphical Database for Category Theory:

<b>CTRL+C</b>	Create a Category
<b>CTRL+H</b>	Display Help File for GDCT
<b>CTRL+F</b>	Open a FGL Functor File
<b>CTRL+G</b>	Save a FGL Functor File
<b>CTRL+D</b>	Download a CGL Category File
<b>CTRL+O</b>	Open a CGL Category File
<b>CTRL+S</b>	Save a CGL Category File

### APPENDIX B: MISC. INFORMATION

---

#### Version Information

The Graphical Database for Category Theory is a research project at Mount Allison University that now includes over 80 Java source files and over 30000 lines of code. Version 1.0 of the GDCT was released as a preliminary version which provides the basic functionality and demonstrates the abilities of the final version of the application. Currently, the list of known problems is being dealt with.

#### **Version 1.0: Released March 20, 2000**

This is the preliminary version of Graphical Database for Category Theory The release information for this version is available at <http://mathcs.mta.ca/research/rosebrugh/gdct/v1release.htm>.

#### **Version 1.1: Released Summer 2000**

This version of GDCT contained the following fixes and enhancements:

- The help file system has been rewritten and now includes a complete set of help files which will better enable the user to use the software
- Contact information in the bug report dialog has been updated
- Upon closing the application, a dialog window now prompts the user to see if they want to save changes in modified categories. This dialog window also tells the user what modifications were made so that they can more accurately judge if they want to save these changes.
- The display of categories in the "Make Confluent Tool was changed so that a category isn't redrawn randomly. Instead, GDCT will only update the label of the node containing the list of relations.
- In "Add Data" for categories, the category is no longer redrawn randomly. New data is now added to the existing graphical representation as opposed to creating a new random representation.
- In "Remove Data" the problem where not all data is removed in some categories has been fixed Also, in "Remove Data" for categories, the text display is updated and the graphical display is updated as data is removed.
- In "View GML", the problem with the search feature not working right away has been corrected.
- The "Make Confluent" algorithm has been checked extensively and all known bugs have been fixed.
- Bugs in "Make Dual Category" have been fixed and the make display of a newly created dual category is no longer determined randomly. Instead, the display is based on the display of the category that was originally used to create the dual category.
- The "Equality of Composites" tool has been tested and all problems have been fixed.
- In the "Initial Object" and "Terminal Object" tools a warning is now displayed if the endomorphism limit has been reached.

- GDCT 1.0 was developed using Borland JBuilder 2.0 and used some borland libraries such as `borland.jbcl.control` and `borland.jbcl.layout`. This caused certain compiling problems in non-Borland environments. To correct this situation, all elements of borland libraries have been removed.

### Version 2.0: Released May 2002

This version of GDCT contained the following fixes and enhancements:

- Diagram display added
- When in diagram display mode, the "View GML" tool will show the GML for the diagram category
- Open for \*.FGL files implemented
- Download for \*.FGL files implemented
- Added "Functor Graphical Settings" items to the initialization file
- Check equalizer/coequalizer tools added
- Check monomorphism/epimorphism tools added
- Check isomorphism tool added
- Generalized "Sum" and "Product" algorithms
- In "Functor Graphical Settings", fixed the "AB Vertical", "A Only", and "B Only" display modes, and disabled the "AB Horizontal" display mode due to compatibility problems
- Fixed sum and product algorithms
- Fixed functor animation problems
- Fixed problems when switching between functors and categories using the pulldown boxes
- Fixed problem where the category GML would be lost when adding relations to a category
- Fixed problem where \*.FGL and \*.FUN files were not always updated in the Recent Files list
- Fixed problems when switching between functor display and category display mode, the GML would sometimes be lost
- Fixed bug where changes to category graph were lost when opening dialog boxes
- Fixed bug when trying to close categories and functors
- Many other small bug fixes

### Known Bugs

Below is a list of known bugs in the current version of the GDCT application. If you have encountered additional bugs please fill out a comment form at <http://mathcs.mta.ca/research/rosebrugh/gdct/contact.htm>.

- Problem with "A B Horizontal" view for functors. Bug sidestepped by disabling this option.

### Proposed Enhancements

Below is a list of enhancements that will be made to the GDCT application. If you have ideas for additional enhancements please fill out a comment form at <http://mathcs.mta.ca/research/rosebrugh/gdct/contact.htm>.

- "Pullback" tool
- The ability to use tools such as "Sum", "Equaliser", "Isomorphism", in functors

