



How to Use the GBB Graphics System with the Limo Example

GBB Version 4.1

Overview

What this document is about

This document gives you a step-by-step guide to using the features of the GBB™ Graphics System. The exercises are based on the GBB limo application, which is a demonstration application that acts as an intelligent dispatcher in scheduling limousine service for customers traveling between various locations.

This document describes the procedures you use to perform activities such as:

- Activating the graphics system and starting an application
- Creating blackboard windows and tailoring them to display the information you need
- Using the Control Shell Window

You should begin by loading and running the limo application, as described in "**Loading and running the limo application.**" Then, follow the instructions for creating windows and selecting space instances and dimensions for the windows. After you've created the windows, you can enter orders for limos and watch the application record the orders and dispatch limos to service the customers.

Mouse use

The GBB Graphics System is a graphics interface created using the ChalkBox™ graphics toolkit. Because ChalkBox runs on a variety of window toolkits, it uses a machine-independent mouse model that is based on a three-button mouse and six types of clicks: a single click of the left, middle, or right mouse button, or a shift-left, shift-middle, or shift-right action (that is, pressing the Shift button while clicking the mouse button). Some window toolkits directly support these six types of clicks; others don't.

For information on how to use your mouse with ChalkBox and how to use ChalkBox's pop-up menus and dialogs, see the *Interacting with ChalkBox* manual. It describes the keystroke/mouse-button combinations that simulate the different mouse-button clicks in the window toolkits that don't directly support ChalkBox's six types of clicks.

Tip: The GBB mouse documentation displayed in a window at the lower left corner of your screen provides a handy source of information on the functions of your mouse buttons or keystroke/mouse-button combinations.

For more information

See the following documents for more information:

- **About GBB**—To learn more about the GBB concepts discussed in this tutorial, refer to the GBB documentation set.
- **About the limo application**—To learn more about the limo application, see the following documents:
 - *The Limo Example*—A descriptive document that provides (1) general background information about the limo application and (2) detailed information about the spaces, unit classes, dimension-value computations, and knowledge sources (KSs) used by the limo application.
 - *How to Use the Limo Example*—A tutorial describing how to use the limo application (for example, how to enter orders and inspect unit instances). It specifies how to run the limo application so that it uses a predefined window configuration (by simply specifying `(limo-example t)` to start the application).

This tutorial differs from *How to Use the Limo Example* in that it describes the individual steps for creating and tailoring the various blackboard windows; creating the Control Shell Window; and performing various graphics operations. However, the graphics windows you create using this tutorial closely resemble those that are displayed automatically when you run the limo example as specified in *How to Use the Limo Example*.

Loading and running the limo application

Loading start-up information

In the Lisp listener window, make the directory containing GBB current and load the `startup.lisp` file into Lisp by entering the following form:

```
(load "startup.lisp")
```

Lisp loads the `startup.lisp` file and several other files.

Tip: You must specify a complete path name if the `startup.lisp` file is stored in a directory other than your default directory. For example, the form might be:
`(load "/local/gbb/v-410/startup.lisp")`.

Switching from another GBB application

If you are switching to the limo example from any other GBB application (for example, the ecosystem example or elevator example), simply reset GBB by calling the **reset-gbb** generic function, instead of loading the `startup.lisp` file.

Loading and running the limo example

To load the GBB `limo-example` module and run the application, enter forms in the Lisp listener window as follows:

- 1 Load the `limo-example` module:

```
(load-kti-module :limo-example)
```

- 2 Change to the `limo-example` package:

```
(in-package :limo-example)
```

- 3 Activate the GBB Graphics System:

```
(initialize-chalkbox)
```

The GBB logo window appears in the upper, right corner of the screen and the ChalkBox Mouse Documentation window appears in the lower left corner of the screen.

- 4 Run the `limo-example` module:

```
(limo-example)
```

Note: If your application window is not sufficiently large or your monitor does not have sufficient resolution, a Chain Manager window will be displayed, to allow you to iteratively select among individual windows. For information about using the Chain Manager window, see "**Small screen displays**" in the document titled *How to Use the Limo Example*.

The next step

When you run the `limo-example` module, GBB runs the limo task. Notice the trace lines that are displayed in the Lisp listener window, indicating the events signalled by the limo task. Then, the application hibernates while awaiting a limo order.

At this point, before entering a limo order, move the Lisp listener window to the lower right corner of your screen and then proceed to the steps described below for creating blackboard windows. After you create the blackboard windows in which application data will be displayed, this tutorial will instruct you to enter orders and monitor the resulting activities.

Building a suspended Lisp image

Note that Lisp loads all of the required GBB files each time you load the `limo-example` module. Therefore, to avoid reloading all those files each time you want to run the limo example (or another GBB example), you can

build a suspended Lisp image containing the commonly used GBB modules. Then, when you load the `:limo-example` module to run the limo example, only the files specific to the limo example are loaded.

For example, the following forms build an image containing the `gbb`, `gbb-graphics`, and `agenda-shell` modules:

```
(load-kti-module '(:gbb :agenda-shell :gbb-graphics))

(setq excl::*read-init-files* nil)

(kti-tools:save-image "my-gbb")
```

Recovering from errors

If you cause an error while performing the activities in this tutorial, see "[Error recovery](#)" at the end of this document. That section describes how to recover from the most common errors.

Creating blackboard windows

The steps in this section will help you create the five blackboard windows used by the limo example (that is, as used by the predefined version of the limo example described in the *[How to Use the Limo Example](#)* document). The windows will show the following information:

- A route map for Lazy Limo Co.
- The limo orders received at Lazy Limo Co. Each time you enter a limo order, the window will display a line segment between the origin and destination places, showing the shortest route between the two places as the crow flies.
- The planned routes for servicing limo orders. Each time you enter a limo order, the window will display a line segment or a series of line segments between the origin and destination places, showing the fastest route between the two places, given other constraints.
- A graph showing a time/place view of the limos. A graph showing the status of the limos (for example, waiting or carrying fare).

Remember: When performing the following steps, refer to the *Interacting with ChalkBox* manual for help in using menus and your mouse (for example, to find out how to close a menu or determine whether you need to use a keystroke/mouse-button combination to simulate a particular type of mouse-button click.)

Creating the Route Map window

Place and road unit instances

The Route Map window will display a map showing the place and road unit instances stored on the x and y dimensions of the map space instance. Perform the following steps to create the Route Map window and display unit instances:

- 1 Bring up the ChalkBox Application Menu by clicking left on the GBB logo window
- 2 Create a blackboard window by clicking appropriately on **Create a Blackboard Window** on the ChalkBox Application Menu. GBB displays a blackboard window titled No Current Space, indicating that you haven't yet selected a space instance for which unit instances are to be displayed.

Note: The type of mouse click you use to select an operation from a pop-up menu depends on your window toolkit. The mouse documentation indicates which mouse click you should use. (Also, see your window toolkit's documentation for more information.)

- 1 Select the space instance and dimensions for which unit instances are to be displayed, as follows:
 - a Move the mouse cursor onto the center of the blackboard window and click right to bring up the Graph Operations Menu.
 - b Click appropriately on **Choose Spaces**. GBB displays the Spaces and Dimensions Menu, which contains a list of all space instances in the blackboard database.
 - c Select the map space instance by clicking left on **map** on the menu of space instances. GBB then displays a list of all dimensions that have been defined for the map space instance. Note that GBB also indicates the number of unit instances on the selected space.
 - d Choose x and y for the dimensions to be displayed for the map space instance by clicking left on **x** for the x-axis dimension and on **y** for the y-axis dimension.
 - e To specify the label that is to be displayed at the top of the window, enter the string **"Route Map."**
 - f Click left on the OK button.

Tip: Alternatively, you can bring up the Spaces and Dimensions Menu by simply clicking left on the center of the blackboard window.

In the blackboard window, GBB displays a map that shows the place and road unit instances located on the x and y dimensions of the map space instance. (Note that the places at which a sign is displayed are designated as waiting places for the limos.)

Creating the Orders Received window

Place and order unit instances

The Orders Received window will display a map showing the place and order unit instances stored on the x and y dimensions of the map and orders space instances. Perform the following steps to create the Order Received window:

Note: Until you enter an order, the window will display only place unit instances.

- 1 Bring up the ChalkBox Application Menu again by clicking left on the GBB logo window.
- 2 Create a blackboard window by clicking appropriately on **Create a Blackboard Window** on the ChalkBox Application Menu. GBB displays an empty window.
- 3 Move the new window from on top of the first blackboard window to another location on the screen, by using the technique provided by your window system. Alternatively, you can use the following method:
 - a Bring up the Window Operations Menu by using clicking shift-right on the new blackboard window in the area outside the empty box and the title area.
 - b Select **Move Window** from the Window Operations Menu.
 - c Position the mouse cursor on the point at which you want the upper, left corner of the window to be placed and click left.
- 4 Select the space instances and dimensions for which unit instances are to be displayed, as follows:
 - a Bring up the Spaces and Dimensions Menu by simply clicking left on the center of the new blackboard window.
 - b Associate the orders and map space instances with the window, by clicking left consecutively on **orders** and **map**. GBB then displays a list of dimensions that are common to the orders and map space instances. (Note that x and y are the only dimensions that are common to both space instances.)
 - c Choose x and y for the dimensions to be displayed for the orders and map space instances, by clicking left on **x** for the x-axis dimension and on **y** for the y-axis dimension.
 - d To specify the label that is to be displayed at the top of the window, enter the string **“Orders Received.”**
 - e Click left on the OK button.

GBB then displays a map that looks the same as the one in the Route Map window. The map currently looks like the one in the Route Map window because you haven't yet entered a limo order. Also, in this window you don't need to see the roads serviced by the Lazy Limo Co., so you can remove them from the display by unselecting them. (Since both place

and road unit instances are stored on the map space instance, both are displayed by default.)

- 5 Unselect the road unit class as follows:
 - a Click right on the graph in the Orders Received window to bring up the Graph Operations Menu.
 - b Invoke the Choose Unit Classes operation in order to display a menu of unit classes.
 - c Click left on **road** and then on the OK button, to unselect the road unit class.

Tip: Alternatively, you can bring up the Unit Classes Menu by clicking shift-left in the margin area of the blackboard window.

GBB redisplay the map in the window; this time it displays the places, but not the roads between places.

Changing the color of orders

Next, change the color of the order unit class in order to make orders easily distinguishable. To change colors in the limo application, you use the Unit Colors Menu as described below.

Bringing up the Unit Colors Menu

You bring up the Unit Colors Menu from the particular window containing the unit class or classes for which you want to change colors (in this case, the Orders Received window). From the window, use either of the following methods:

- 1 Click right anywhere on the graph to bring up the Graph Operations Menu. Then click appropriately on **Choose Unit Colors**.
- 2 Click shift-middle in the margin area of the graph.

The Unit Colors Menu contains a menu of unit classes on the left, a menu of foreground colors in the middle, and a menu of fill colors on the right. Notice that the foreground and fill colors for each unit class are black by default.

Changing a unit class's color

Use the Unit Colors Menu to change the color of the `order` unit class as follows:

- 1 Click left on **order** to select it and then click left on **cyan** in the foreground colors menu. (The fill color doesn't apply to lines, so you need not select a fill color.)
- 2 Click left on the Set button to set the foreground color of the `order` class to the selected color.
- 3 Click left on the Apply button and then click left on the OK button to exit from the Unit Colors Menu.

Then, when you enter an order, it will be drawn in cyan in the Orders Received window.

Creating the Planned Tasks window

Place, empty, and carrying-fare unit instances

The Planned Tasks window will display a map showing the place, empty, and carrying-fare unit instances stored on the x and y dimensions of the map and tasks space instances. Perform the following steps to create the Planned Tasks window:

Note: Like the Orders Received window, the Planned Tasks window will display only place unit instances, until you enter an order.

- 1 Bring up the ChalkBox Application Menu by clicking left on the GBB logo window.
- 2 Create a blackboard window by clicking appropriately on **Create a Blackboard Window** on the ChalkBox Application Menu.
- 3 Move the new window from on top of the first blackboard window to another location on the screen. (You can move and reshape the blackboard windows as necessary to fit them all on your screen.)
- 4 Select the space instances and dimensions for which unit instances are to be displayed, as follows:
 - Bring up the Spaces and Dimensions Menu by simply clicking left on the center of the blackboard window.
 - Click left consecutively on **tasks** and **map**.
 - Click left on **x** for the x-axis dimension and on **y** for the y-axis dimension.
 - To specify the label that is to be displayed at the top of the window, enter the string **“Planned Tasks.”**
 - Click left on the **OK** button.

Unselect the road unit class, since you needn't see the roads in this window. Use the same procedure you used when unselecting the unit class from the Orders Received window. GBB redisplay the map, showing the places but not the roads between places.

Changing the color of roads, and empty and carrying-fare tasks

Change the color of roads and empty and carrying-fare tasks to allow you to more easily distinguish them in the blackboard windows. As a result, the roads shown in the Route Map window will immediately be redisplayed in yellow. Further, after you enter a limo order, you'll be able to easily distinguish the carrying-fare task (representing the application's proposed route for the limo, from the place of origin to the destination place) from the empty task (representing the route the limo must take to get from its initial waiting place, Lazy Limo Co., to the place of origin).

To select the new colors, perform the following steps:

- 1 **Bring up the Unit Colors Menu** from the Planned Tasks window. By invoking the color changes from the Planned Tasks window, you can

change all three unit classes at once, since the road unit class is stored on the map space instance and the empty and carrying-fare unit classes are stored on the tasks space instance, both of which are selected for the Planned Tasks window.

- 2 Click left on **road** to select it and then click left on **yellow** in the foreground colors menu. Then, click left on the **Set** button.
- 3 Click left on **carrying-fare** to select it and then click left on **red** in the foreground colors menu. Then, click left on the **Set** button.
- 4 Click left on **empty** to select it and then click left on **green** in the foreground colors menu. Then, click left on the **Set** button.
- 5 Click left on **All widgets**, thereby indicating that GBB is to set the color of the carrying-fare and empty unit classes (as well as all other unit classes listed in the Unit Colors Menu) in its global unit/color table.
- 6 Click left on the **Apply** button to apply the color selections and then click on the **OK** button to exit from the Unit Colors Menu.

As a result, the new colors will immediately be shown in existing blackboard windows that display carrying-fare or empty instances. Further, the next two blackboard windows you create will display instances of these unit classes in the new colors.

Notice that in the Planned Tasks window the dot at Lazy Limo Co. is now green. The green color indicates instances of the empty unit class, which represent the waiting activity of all the limos at the Lazy Limo Co.

Creating the Time/Place View window

Empty and carrying-fare unit instances

The Time/Place View window will display a graph showing the empty and carrying-fare unit instances stored on the time and place dimensions of the tasks space instance. Perform the following steps to create the Time/Place View window:

- 1 Bring up the ChalkBox Application Menu by clicking left on the GBB logo window.
- 2 Create a blackboard window by clicking appropriately on **Create a Blackboard Window** on the ChalkBox Application Menu.
- 3 Move the new window from on top of the first blackboard window to another location on the screen.
- 4 Select the space instances and dimensions for which unit instances are to be displayed. Select the **tasks** space instance; select **time** for the x-axis dimension and **place** for the y-axis dimension; and specify **“Time/Place View”** for the label of the window.

GBB then displays a graph showing that all limos are at the Lazy Limo Co. (since you haven't yet entered a limo order) and are empty (as indicated by the green color).

Creating the Task Status window

Empty and carrying-fare unit instances

The Task Status window will display a graph showing the empty and carrying-fare unit instances stored on the `time-interval` and `task-status` dimensions of the `tasks` space instance. Perform the following steps to create the Time/Place View window:

- 1 Bring up the ChalkBox Application Menu by clicking left on the GBB logo window.
- 2 Create a blackboard window by clicking appropriately on **Create a Blackboard Window** on the ChalkBox Application Menu.
- 3 Move the new window from on top of the first blackboard window to another location on the screen.
- 4 Select the space instances and dimensions for which unit instances are to be displayed. Select the `tasks` space instance; select `time-interval` for the x-axis dimension and `task-status` for the y-axis dimension; and specify “**Task Status**” for the label of the window.

GBB then displays a graph showing that all limos are waiting at the Lazy Limo Co. (as opposed to being en route to pick up a fare or travel to a waiting place) and are empty (as indicated by the green color).

Entering a limo order

In the Lisp listener window

Now, you can enter a limo order and watch the information in the blackboard windows change. Notice the route of the assigned limo as it is displayed in the Planned Tasks window. (To understand how the application determined the planned route, check the possible routes in the Route Map window.) If the Lisp listener window is hidden by other windows, bring it to the front of the windows. If the cursor is not at the `Origin?` prompt, press Return to display the `Origin?` or `Enter a place name` prompt.

Perform the following steps to enter an order:

- 1 At the `Origin?` or `Enter a place name` prompt, enter **j-mart** for the name of the place of origin for the limo (that is, the place at which the customer wants to be picked up), and press Return.
- 2 At the `Destination?` prompt, enter **golden** (which is short for Golden Age Village) for the name of the destination place (that is, the place to which the customer wants to be delivered), and press Return.
- 3 At the `Pickup time [hh:mm]?` prompt, enter **10:00**, which indicates a pickup time of 10 A.M., and press Return.
- 4 At the `Number of passengers?` prompt, enter **2** and press Return.

KS activity

On the basis of the limo order you entered, GBB runs KSs that determine:

- Which limo is available to pick up two passengers at J-Mart, at 10 A.M., and deliver them to Golden Age Village
- The fastest and most efficient route for servicing these customers, given the possible routes (as defined to the application)

As GBB runs the KSs, it displays trace information in the Lisp listener window. Note the message displayed by one of the KSs:

```
Assigning limo n to pickup 2 passengers at 10:00 from
J-Mart. Drop off is at 11:24 at Golden Age Village.
```

Before you enter more orders, create the Control Shell Window so you can monitor the activities of the control shell as it oversees the processing of KSs.

Creating the Control Shell Window

Monitoring control-shell activities

Create the Control Shell Window as follows:

- 1 Bring up the ChalkBox Application Menu, by clicking left on the GBB logo window.
- 2 Click appropriately on Create Control Shell Window.

GBB displays the Control Shell Window. You may need to resize some or all of the windows on your screen (including the Control Shell Window) in order to see all the windows.

Next, enter more limo orders (as described below) and watch the Control Shell Window to see the resulting processing activities.

Entering additional limo orders

Ad hoc orders

If the Lisp listener window is hidden by other windows, bring it to the front of the windows. If the cursor is not at the `Origin?` prompt, press `Return` to display the `Origin?` or `Enter a place name` prompt.

Perform the following steps to enter an order:

- 1 At the `Origin?` or `Enter a place name` prompt, enter `?` and press `Return`. GBB responds by displaying a list of all place names.
- 2 At the `Enter a place name` prompt, enter a place name for the origin place and press `Return`. (Note that you can enter only the first few letters or characters of the place name.)
- 3 At the `Destination?` prompt, enter a place name for the destination place and press `Return`.
- 4 At the `Pickup time [hh:mm]?` prompt, enter a time and press `Return`.

5 At the Number of passengers? prompt, enter a number in the range 1–4 and press Return.

Next, enter some additional orders of your choice.

Inspecting unit instances

Slots and dimension values

You can explore the unit instances in the blackboard windows by using GBB's Find Units in Region and Find Units Near Point operations. Also, try using the Zoom In and Zoom Out operations. Use the mouse documentation to guide you.

Note: The mouse cursor must be over the graph portion of a blackboard window in order for you to directly access the **Find Units** and **Zoom** operations.

Refer to the section titled "**Exploring unit instances**" in the *How to Use the Limo Example* document for specific instructions on inspecting unit instances.

Using other graphics operations

Numerous choices

The GBB Graphics System allows you to perform a variety of operations on both the data displayed in a window (for example, changing the bounds of the dimensions for which unit instances are displayed or filtering the unit instances to be displayed) and the window itself (for example, changing the foreground color or the font).

These operations are available from menus, which group the operations as follows:

- Graph Operations Menu
- Widget Operations Menu
- Window Operations Menu

Refer to the mouse documentation to determine how to access these menus. (The Window Operations Menu is accessible from the Widget Operations Menu.)

Note: To experiment with these operations, you can create additional blackboard windows or use the blackboard windows you've already created.

Quitting the application

To quit the limo example, enter `:q` at the Origin? prompt in the Lisp Listener window and press Return.

Error recovery

When using Allegro Common Lisp

If you're running the limo application on Allegro Common Lisp and either of the following conditions apply, the problem described below can occur:

- You're running Lisp on UNIX and not using the emacs interface
- You're running Lisp on Windows and not using either the emacs interface or Allegro CL's IDE

When an error is signalled, the error process and the normal Lisp listener process are both trying to read from the same stream. (More specifically, the debugger and the limo example are trying to read input at the same time, and they succeed in reading only alternate lines.)

To exit from an error, enter `:pop` twice in the Lisp listener window to ensure you're really out of the error.

For more information

About GBB and ChalkBox

For more information about the GBB concepts introduced in this tutorial, refer to the GBB documentation set. For more information about the ChalkBox graphics toolkit (on which the GBB Graphics System was developed), refer to the ChalkBox documentation set.

Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Knowledge Technologies International, Inc.

U.S. Government Restricted Rights Legend: Use, duplication, and disclosure by the Government are subject to restrictions of Restricted Rights for Commercial Software developed at private expense as specified in DOD FAR 52.227-7013 (c)(1)(ii).

Copyright © 2000 by Knowledge Technologies International, Inc. All rights reserved.

GBB, ChalkBox, KTI Tools, and the KTI logo are trademarks of Knowledge Technologies International, Inc. Other brand or product names are trademarks or registered trademarks of their respective holders.

May 2000

Knowledge Technologies International, Inc.
401 Main Street
Amherst, MA 01002
Phone: (800) 577-8990, (413) 256-8990
Fax: (413) 256-3179
E-mail: gbb-info@KTIworld.com
WWW: www.KTIworld.com



KNOWLEDGE
TECHNOLOGIES
INTERNATIONAL

www.KTIworld.com