V12 Database Engine® Behaviors Library

version 1.0

for Macromedia Director® 6.x

(98/06/06)

integration
new media

©Integration New Media, Inc. 1998

Table of Contents

©Integration New Media, Inc. 1998

1. Introduction

The V12-DBE Behaviors Library enables you to implement V12's functionalities into your projects without the need of writing Lingo scripts or even naming Director members. You perform this by dragging and dropping behaviors over user interface elements and filling in parameters in dialog boxes.

You need very little background with V12 Database Engine to develop your own projects with the V12-DBE Behaviors Library. Before you start using V12-DBE behaviors, we recommend that you acquire some notions of databases and V12-DBE. You mainly need to know what a *database*, a *table*, a *field*, a *current record* and a *selection* are. You can acquire those notions by:

- viewing the V12 Database Engine Tutorial,
- reading the Database Basics section of the V12-DBE user manual,
- skimming through the Using V12-DBE section of the V12-DBE user manual.

V12-DBE Behaviors were designed to allow for the fast development of database-driven user interfaces. They are versatile and easy to use but do not provide for all the functionalities one can implement when using V12-DBE directly from Lingo. See V12-DBE Behaviors versus Lingo Scripts.

If you never used Director behaviors before, you may want to read Mixing V12-DBE Behaviors and Lingo scripts before reading on.

If you are already familiar with behaviors or hate reading manuals, go directly to V12-DBE behaviors step-by-step for a first experiment with V12-DBE behaviors. You may eventually want to read this entire manual after your first round of practice.

In either cases, make sure that you understand Director 6's Cancel button bug mentioned on page 20.

The V12-DBE Behaviors Library is free to use and redistribute. When redistributing the V12-DBE Behaviors Library (whether via Internet, CD-ROM, personally or otherwise), you must make sure that they are unmodified from their original version. Integration New Media Inc. withholds all rights to the V12-DBE Behaviors Library.

2. Categories of V12-DBE Behaviors

V12-DBE behaviors are in three categories:

- Action Behaviors
- Data Managers
- Messengers

2.1. Action Behaviors

Action Behaviors are generally attached to buttons or equivalent "clickable" user interface elements. They perform V12-DBE actions, such as browsing through data, searching, importing/exporting files, etc.

Since they are dependant upon "clickable" user interface elements, they are triggered by MouseUp or MouseDown events. In addition to performing V12-DBE-specific actions, Action Behaviors can ask Data Managers to save or display data. They are characterized by the presence of an "Event" pop-up menu and either (or both) of the "Update before action" and "Refresh after action" checkboxes.

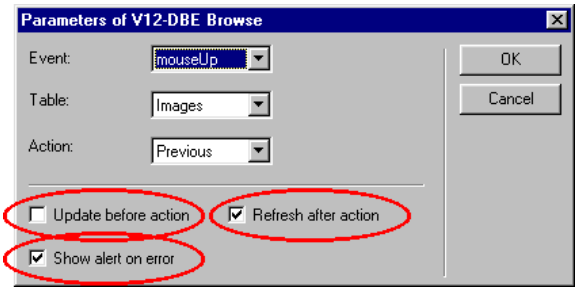


Figure 1: Example of an Action Behavior.

The V12-DBE Behaviors Library contains the following Action Behaviors (see Using V12-DBE Behaviors for detailed explanations):

- Open Database
- Close Database
- Browse
- Go to Record
- Search All
- Search with one criterion
- Search with two criteria
- Search with three criteria
- Add Record
- Delete Record
- Delete Selection
- Import Text File
- Import DBF File
- Import V12-DBE File
- Export Text File
- Export DBF File

©Integration New Media, Inc. 1998

2.2. Data Managers

Data Managers are generally attached to Field members. Each Data Manager is responsible for displaying content originating from V12-DBE into the Director member it is attached to, or saving data from that Director member to V12-DBE. Data Managers can also be attached to bitmap members to display and save Media to V12-DBE fields.

A Data Manager displays data from V12-DBE when it receives a V12Refresh message. Such a message is broadcasted to all Data Managers by Action Behaviors who's "Refresh after action" box is checked. They can also be explicitly sent by "Refresh Stage" Messengers (see below).

A Data Manager saves data from the Director member it is attached to, to V12-DBE when it receives a V12Update message. Such a message is broadcasted to all Data Managers by Action Behaviors who's "Update before action" box is checked. They can also be explicitly sent by "Update Record" Messengers (see below).

The V12-DBE Behaviors Library contains the following Data Managers (see Using V12-DBE Behaviors for detailed explanations):

- Display/Save Data
- Display Selection
- Table Info
- Database Info

Future versions of the V12-DBE Behaviors Library may also implement custom messages, to be used when refreshing or updating data partially.

2.3. Messengers

Messengers are responsible for sending Refresh or Update messages to the Data Managers. They are convenient to attach to buttons that refresh fields or update data without performing V12-DBE actions. They are less frequently used than the two other types of behaviors.

The V12-DBE Behaviors Library contains the following two messengers:

- Update Record
- Refresh Stage

©Integration New Media, Inc. 1998

3. Installing "V12-DBE Behavior Lib.cst"

The V12-DBE Behaviors Library is a self-contained Director CastLib named "V12-DBE Behaviors Lib.cst". It does not require any special installation procedure.

However, for your convenience, you may want to drag it to the Xtras folder located in the same folder as Director 6.x and restart Director. Once Director opens, you should see "V12-DBE Behaviors Lib" appear in the Xtras menu.

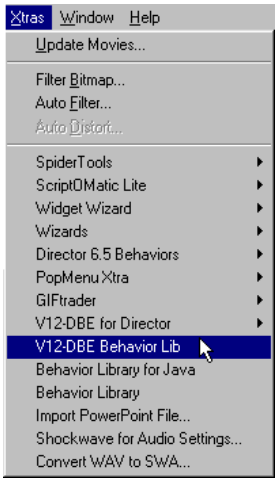


Figure 2: V12-DBE Behaviors Lib once dragged to the Xtras folder.

If you choose not to install "V12-DBE Behaviors Lib.cst" in Director's Xtras menu, then you can open it directly from the File > Open menu item.

©Integration New Media, Inc. 1998

4. V12-DBE behaviors step-by-step

V12-DBE behaviors are designed to help you quickly build data management interfaces including functionalities such as viewing, modifying, deleting, importing records, etc. However, they are not meant for creating new database files. New V12-DBE files can be created through simple scripts as demonstrated in the V12-DBE QuickStart. Alternatively, you can use one of the many database structures supplied with the V12-DBE Behaviors Library, in the Templates folder.

For your first experience with V12-DBE behaviors, proceed as follows:

1. Open the Portfolio.DIR movie from the Step-by-Step folder.

This movie contains a totally inert user interface: just cast members placed in Frames 1 to 28 and no scripts at all, except for the Pause on ExitFrame.

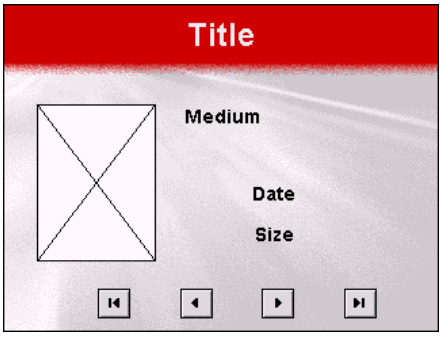
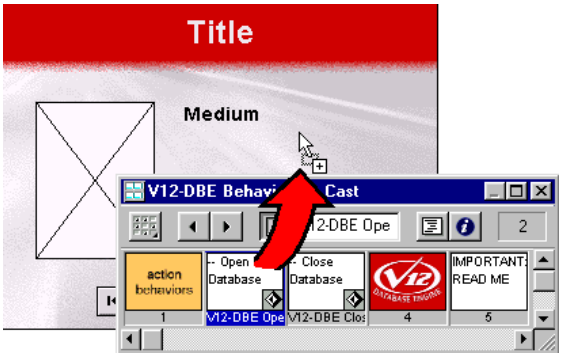


Figure 3: Stage of the Portfolio.DIR movie.

2. Choose the Xtras > V12-DBE Behaviors Lib menu item.

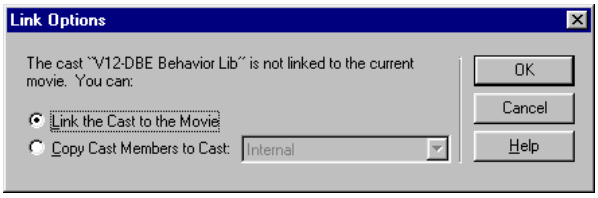
The V12-DBE Behaviors Library opens.

3. Drag the "Open Database" behavior to the background image.



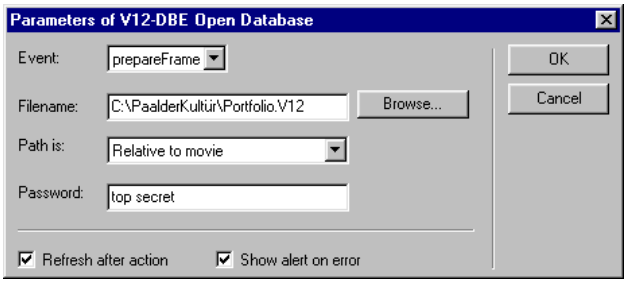
©Integration New Media, Inc. 1998

The first time you use a behavior from the V12-DBE Behavior Library, you are prompted to specify whether you wish to link the Castlib to your current movie or copy behaviors to your own Castlibs.

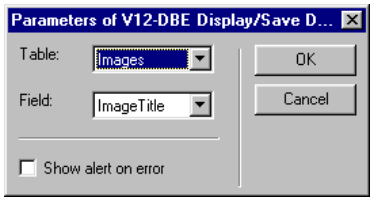


4. Choose "Link the Cast to the Movie".

5. In the parameters dialog box select the Event "PrepareFrame", click the Browse button and pick the "Portfolio.V12" file and type "top secret" in the Password box. Make sure "Refresh after action" is checked.



6. Drag the "Display/Save Data" behavior to the "Title" field on the stage. In the behavior's parameters dialog box, select the Table "Images" and the Field "ImageTitle".

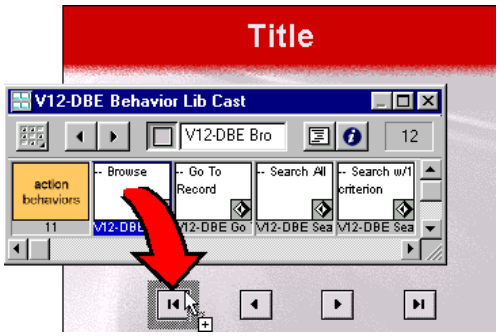


7. Drag the "Display/Save Data" behavior to each of the three other fields and select Fields "Medium", "DesignDate" and "Size" respectively in each of the parameters dialog boxes.

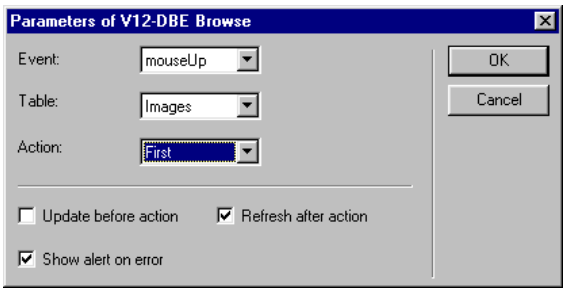
8. Drag the "Display/Save Data" behavior to the Bitmap sprite named "Image". In the behavior's parameters dialog box, select the Table "Images" and the Field "Image".

9. Drag the Browser behavior to the "First" button.

©Integration New Media, Inc. 1998



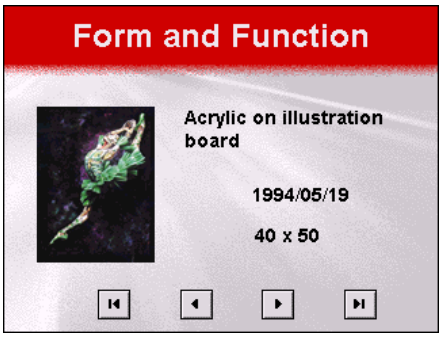
10. In the parameters dialog box, select the Event "MouseUp", the Table "Images" and the Action "First". Make sure the "Refresh after action" and "Show alert on error" boxes are checked. Since you will be only viewing records (not modifying them) in this example, you can ignore the "Update before action checkbox.



11. Repeat step 10 for the Previous, Next and Last buttons. You need to select the Actions "First, "Next" and "Last" in each of these cases.

12. Run the movie.

You should immediately see the first record load to your sprites. Click successively on the Next, Previous, First and Last buttons to browse through the records of the "Portfolio.V12" database.



Congratulations! You just completed your first database browsing user interface without writing a single line of Lingo.

There is much more you can do using V12-DBE Behaviors than this simple record browsing interface: record updating, keyword search, selection in lists, and much more… are features that you can very easily add to this project.

4.1. What is actually happening under the hood

To help you better understand the V12-DBE behaviors' underlying mechanism, this is the sequence of operations performed by the above example:

- On PrepareFrame, the Open Database behavior is automatically triggered by Director and opens the "Portfolio.V12" database. Since the "Refresh after action" box is checked in the Open Database behavior, a "V12Refresh" message is broadcasted to all the sprites currently on stage.

- Data Messengers are triggered by the "V12Refresh" message. Each one of them retrieves the V12-DBE field it is assigned to, to the container it is attached to (whether a field or a bitmap).

- When the user clicks on the "Next" button, a "MouseUp" message is sent to it and the Browser behavior that is attached to it (and configured to react to "MouseUp" events) is triggered. It performs an mGoNext operation on the specified V12-DBE table (thus moving the current record to the next one) and then broadcasts a "V12Refresh" message to all the sprites currently on stage.

- Once again, Data Messengers respond to the "V12Refresh" message and retrieve the content of the V12-DBE fields they were assigned to.

- The same mechanism applies to the other browsing buttons too (First, Previous and Last).

Important:	Each sprite is responsible for displaying its own content. Never does any behavior affect the content of a sprite it is not attached to. This concept is key to understanding the V12-DBE Behaviors Library. It is in fact one of the foundations of OOP (Object-Oriented Programming).

©Integration New Media, Inc. 1998

5. Using V12-DBE Behaviors

V12-DBE Behaviors are explained in details in their own description pane, in the Behavior Inspector's window. They are generally straightforward to use. Following are a few additional guidelines. Unless otherwise specified, the behaviors listed below are all Action Behaviors.

5.1. Open Database

The "Open Database" behavior is key to all the other behaviors. All V12-DBE behaviors rely on it.

For a V12-DBE behavior (whether an Action Behavior, a Data Manager or a Messenger) to properly operate, it needs to access the current V12 database's structure. This is why the "Open Database" behavior uses a few of your own Castlib's members to store data such as the database pathname, its password, etc. That way, any behavior can force the database open to read its structure when needed.

It is generally a good idea to attach the "Open Database" behavior to an invisible sprite on the stage or the background image, and have it respond to the "PrepareFrame" event. That way, you can more easily access its parameters dialog box to view or modify parameters.

Also note that the "StartMovie" movie event is never sent to sprites. Thus, if you want your "Open Database" behavior to trigger on "StartMovie", you must attach it to the frame (not recommended).

5.2. Close Database

For simple projects where you need to open a V12-DBE database on StartMovie and close it on exit, the "Close Database" behavior is not mandatory. V12-DBE and Director (or its Projector) have the ability to collaborate to properly close the current database file. If the end-user is allowed to modify data in the database, it is preferable to close the database explicitly upon exit.

5.3. Get/Set Data

The Get/Set Data behavior is a Data Manager. It is generally attached to Director members of type Field, but they can also be attached to members of type Bitmap (or Sound, or Palette, etc.) to deal with V12-DBE fields of type Media. It uses V12-DBE's mGetField, mGetMedia, mSetField, mSetMedia, mEditRecord, mAddRecord and mUpdateRecord methods.

©Integration New Media, Inc. 1998

To let the user modify a V12-DBE field, set the Editable property of the sprite it is associated to, to TRUE and send V12Update messages to it. Such messages are automatically sent to Get/Set Data behaviors by Action Behaviors whose "Update before action" box is checked.

Note:	Do not create more than a single sprite of each Director field to which Get/Set Data behaviors are attached. If the user is allowed to modify data, such situations can lead to inconsistencies when saving data to V12-DBE.

5.4. Display Selection

The Display Selection behavior is a Data Manager. It retrieves the partial or entire content of the V12-DBE selection to the sprite it is attached to. It retrieves the content of the fields (one or many) selected in the Fields pop-up menu. The "Start at position" and "Number of records" pop-up menus enable you to select the Director fields where the position of the first record to retrieve and the number of records to retrieve can be selected. If you need "hard-coded" values for "Start at position" and "Number of records", you must nonetheless create two Director members to hold those values.

To retrieve the entire selection, check the "Display all records" box or specify a very large number in the "Number of records" field.

The Display Selection behavior is generally attached to a Director member of type Field. It uses V12-DBE's mGetSelection method.

5.5. Browse

The Browse behavior moves the current record to the Previous, Next, First or Last record of the table's selection, depending on the Action pop-up menu's setting.

Checking the "Show alert on error" box displays an error message every time something abnormal happens, including when the user clicks the Next button past the last record of the selection. If such an error checking is inadequate, uncheck the "Show alert on error" box.

The Browse behavior uses V12-DBE's mGoPrevious, mGoNext, mGoFirst and mGoLast methods.

©Integration New Media, Inc. 1998

5.6. Go To Record

The Go To Record behavior moves the current selection to a particular record in the selection specified by its position in the field chosen in the "Position" pop-up menu. It uses V12-DBE's mGo method.

5.7. Search All

The Search All behavior sets the selection to all the records of the designated table, sorted according to the default index or a specific field, as specified by the "Order By" pop-up menu. Only indexed fields can be sorted by the Search All behavior. This is because it relies on V12-DBE's mSelectAll and mOrderBy methods.

5.8. Search with one, two or three criteria

The three behaviors (Search with one, two and three criteria), perform searches in the designated table with the specified search criteria. You can optionally sort the resulting selection via the "Order By" pop-up menu.

The Search behaviors use V12-DBE's mSetCriteria, mOrderBy and mSelect methods.

5.9. Add Record

The Add Record behavior adds a new record to the designated table and broadcasts a message to all the sprites currently on stage to clear themselves. The user can then type the data to store in the new record. Those data are stored to the new record when the sprites that contain them receive as V12Update message. Such a message is sent to them by any Action Behavior whose "Update before action" box is checked.

5.10. Delete Record

The Delete Record behavior deletes the current record.

If the "Ask for confirmation" box is checked, it asks for a confirmation prior to deleting the current record. In that case, the message contained in the member designated by the "Message" pop-up menu is displayed to the user along with the "OK" and "Cancel" buttons. The current record is deleted only if the user clicks on OK.

©Integration New Media, Inc. 1998

Important	You need to have the Mui Dialog Xtra for Director to display a message box with the OK and Cancel buttons. That Xtra is delivered with Director 6.x, in the Xtras folder. If you intend to deliver a project that contains a "Delete Record" behavior with the "Ask for confirmation" option set, make sure to deliver the Mui Dialog Xtra as well. Mui Dialog Xtra is not available for Windows 3.1.

The Delete Record behavior uses V12-DBE's mDeleteRecord method.

5.11. Delete Selection

The Delete Selection behavior deletes the entire selection of the designated table.

If the "Ask for confirmation" box is checked, it asks for a confirmation prior to deleting the records. In that case, the message contained in the member designated by the "Message" pop-up menu is displayed to the user along with the "OK" and "Cancel" buttons. The current record is deleted only if the user clicks on OK.

Important	You need to have the Mui Dialog Xtra for Director to display a message box with the OK and Cancel buttons. That Xtra is delivered with Director 6.x, in the Xtras folder. If you intend to deliver a project that contains a "Delete Record" behavior with the "Ask for confirmation" option set, make sure to deliver the Mui Dialog Xtra as well. Mui Dialog Xtra is not available for Windows 3.1.

The Delete Selection behavior uses V12-DBE's mDeleteSelection method.

5.12. Table Info

Table Info is a Data Messenger. It retrieves the position of the current record (if Info is "Position") or the total number of records in the selection (if Info is "Count") to the sprite it is attached to. It is normally set to respond to the V12Refresh message sent to it by one of the many Action Behaviors. It can also be set to respond to a MouseUp or MouseDown event. This is mostly convenient for debugging purposes.

It is generally not necessary to check the "Show alert on error box".

The Table Info behavior uses V12-DBE's mGetPosition and mSelectCount methods.

©Integration New Media, Inc. 1998

5.13. Database Info

Database Info is a Data Messenger. It retrieves the structure of the current database or the last V12-DBE operation's status or error to the sprite it is attached to. It is normally set to respond to the V12Refresh message sent to it by one of the many Action Behaviors. It can also be set to respond to a MouseUp or MouseDown event. This is mostly convenient for debugging purposes.

The Database Info behavior uses V12-DBE's mDumpStructure, V12Error and V12Status methods.

5.14. Import from Text/DBF/V12 files

The Import behaviors import Text, DBF or V12-DBE files to the current database's designated table. They use V12-DBE's mImportFile method.

Important	If you choose the Path is "Asked at Runtime", you must deliver the FileIO Xtra along with your project to the end-user.

5.15. Export Selection to Text/DBF files

The Export behaviors export the selection to Text or DBF files. Only selected fields can be exported. The sorting order of the resulting file is identical to the selection's order. They use V12-DBE's mExportSelection method.

Important	If you choose the Path is "Asked at Runtime", you must deliver the FileIO Xtra along with your project to the end-user.

5.16. Update Record

The Update Record behavior is a Messenger. It does not perform a V12-related action. It only broadcasts a V12Update message to all the sprites currently on the stage when it receives the event specified in Event.

The Update Record behavior is convenient to attach to a Save button, for example, when you need to force V12-DBE to save the user's input in the current record.

©Integration New Media, Inc. 1998

5.17. Refresh Stage

The Refresh Stage behavior is a Messenger. It does not perform a V12-related action. It only broadcasts a V12Refresh message to all the sprites currently on the stage when it receives the event specified in Event.

Seldom does one need to use the Refresh Stage behavior. It becomes handy when you need to disable the "Refresh after action" option of your Action Behaviors.

©Integration New Media, Inc. 1998

6. Delivering to the end-user

To deliver your project to the end-user, you just need to create a Projector and then simply test it. You do not need to include "V12-DBE Behaviors Lib.cst" into your project.

However, if file size is one of your concerns, you may want to delete members that are useful for authoring only. To do so:

- First make sure you keep an unaltered copy of "V12-DBE Behaviors Lib.cst",
- Delete the behaviors that you never use in your current project from "V12-DBE Behaviors Lib.cst",
- Delete the members "RPD#1" and "RPD#2", which contain Lingo scripts used for authoring only. This approximately saves you 130Kbytes.
- Create the Projector.

©Integration New Media, Inc. 1998

7. V12-DBE Behaviors versus Lingo Scripts

There are benefits and limitations to using V12-DBE behaviors.

Advantages of V12-DBE behaviors

- Easy to use: simply drag and drop behaviors.
- Increased productivity.
- No knowledge of Lingo required.

Limitations of V12-DBE behaviors

- Manage only one V12-DBE database file.
- All tables of the database open up front which provides less control over memory usage.
- No commas in table and field names. It is further recommended that table and field names contain only alphanumeric characters.

7.1. Mixing V12-DBE Behaviors and Lingo scripts

You can mix behaviors and Lingo scripts in a single project without any special consideration if they do not need to interact one with another. For example, if your project's main database is managed by V12-DBE behaviors and you need to add Lingo scripts that deal with another database, no special care is needed: you just need to script the usual way.

If you want to write Lingo scripts that use the same V12 database as the one used by the V12-DBE Behaviors Library, you *must* use the following convenient shortcuts:

- Do not open the database or create table instances yourself. The Open Database behavior properly takes care of opening the database and all the tables contained in it.

- Call V12_GetDBRef() to access the database's Xtra instance. For example, if you need to set the "verbose" property to ON:

```
set x = V12_GetDBRef() -- x is a local variable
mSetProperty(x, "verbose", "on")
```

- Call V12_GetTableRef(*tableName*) to access *tableName*'s Xtra instance. You need to replace *tableName* by the actual name of your table. For example, if you need to automatically bind all members of the Cast "myV12members" to the table "Students" of the current database:

```
set tbl = V12_GetTableRef("Students")
mAutoBinding(tbl, "myV12members")
```

The above guidelines are sufficient to allow for behaviors and Lingo scripts to happily collaborate in a single Director project. Advanced users may want to check the "V12 Entry" member, in "V12-DBE Behaviors Lib.cst" for other convenient handlers.

©Integration New Media, Inc. 1998

Appendix I: Behaviors Basics

Behaviors are special cast members that define operations or procedures. Most behaviors are made to respond to a simple event like a sprite being clicked or the playback head entering a frame. Some behaviors can also respond to events sent to them by other behaviors. Such events are called *messages*.

When the specified event or message occurs, the behavior carries out an action, such as opening a V12 database, retrieving data from a field to a Director member or deleting a record.

Attaching a behavior to a sprite

A behavior is attached to a sprite by dragging and dropping it over that sprite and filling in the parameters in the dialog box.

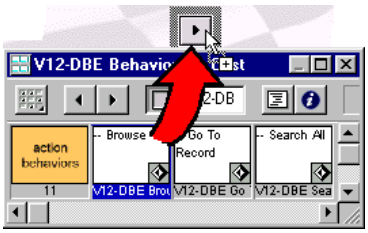


Figure 4: dragging the "Browse" behavior to the "Next" button.

Figure 5 : the "Browse" behavior's parameters.

Click OK to close the dialog box and create a behavior with the specified parameters.

©Integration New Media, Inc. 1998

Important	Clicking Cancel should normally prevent the creation of the new behavior. However certain versions of Director 6 contain a bug and create a dummy behavior even if the Cancel button is clicked. You need to delete those behavior manually. To do so, open the Behavior Inspector window (Windows > Inspectors > Behavior), select the behavior to delete and choose File > Clear Behavior.

You can also modify an existing behavior at any time. To do so, select it in the Behavior Inspector window (Windows > Inspectors > Behavior) and click the Parameters button.

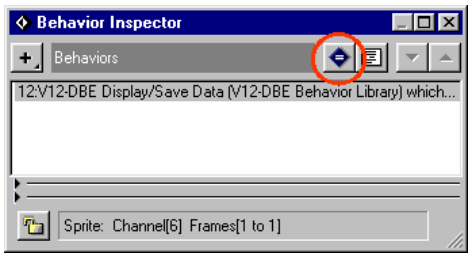


Figure 6: accessing the parameters of a behavior

You can attach as many behaviors as you need to a sprite. Director executes behaviors in the order they were attached to a sprite. You can check that order in the Behavior Inspector window. To change the execution order of a behavior, select it and click the arrows in the Behavior Inspector toolbar to move it up or down.

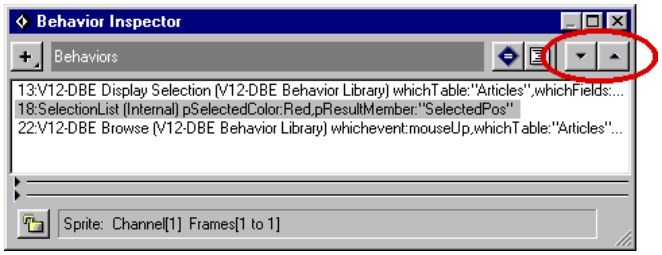


Figure 7: changing the execution order of a sprite's behaviors.

Attaching a behavior to a frame

A behavior can also be attached to a frame by dragging and dropping it over the stage (*not* a background image) or over the score in the Script channel. You can attach only one behavior to a frame.

©Integration New Media, Inc. 1998

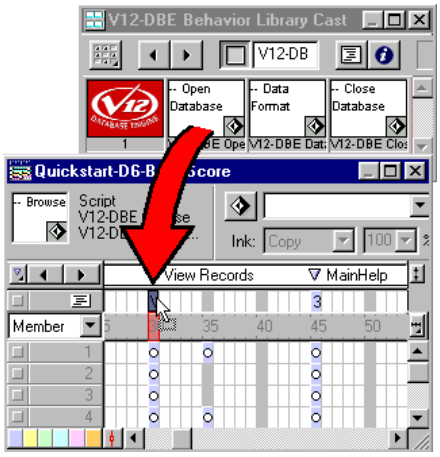
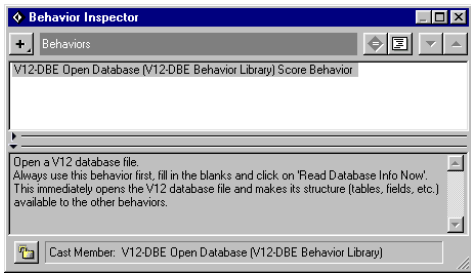


Figure 8: Dragging the "Open Database" behavior to the score script.

Behaviors Descriptions

To see the description of a behavior, open the Behavior Inspector, select a behavior, and click the arrow that expands the Behavior Inspector's description pane. You can leave the description pane expanded and select different behaviors to see their descriptions.



Note:	For Lingo programmers, behaviors are the same as score scripts. They actually contain Lingo scripts, but you never need to see or modify them. If needed, you can address V12-DBE both from behaviors and your own Lingo scripts in a single project. See **Error! Reference source not found.**.

For a more detailed understanding about using Director behaviors see the Director 6 "Using Director" guide, chapter five "Creating Interactivity".

©Integration New Media, Inc. 1998

Acknowledgments

Special thanks to Heidi Taillefer for permitting us to use her work in the Portfolio.DIR movie and to Andrew Newkirk Rose for his invaluable feedback on this project.

©Integration New Media, Inc. 1998