

DataLICENSE

Version 2.0

User Guide

DataLicense is offered as a tool to professional developers of 4th Dimension® databases. It is the responsibility of the purchaser/developer to use this tool in a manner which is not detrimental to the interests of their good-faith clients and customers.

In other words, *if you protect a client's datafile with a serial number* **DO NOT LOSE THE ABILITY TO GENERATE SERIAL NUMBERS TO UNLOCK IT!** Nobody else but you, including Outside Technology, can generate serial numbers for your programs!

LICENSE AGREEMENT

please read

DataLicense is copyright 1995, 1999 Daniel Katz, Outside Technology

All Rights Reserved

The forms, scripts, and methods contained herein, together with the Generator application, are licensed for use by one development entity (company or individual), which may use them in any of its compiled 4th Dimension applications.

They, this User Guide, or any other component of the package with the sole exception of the **compiled** demo, may not be sold, given away, distributed, rented, or transferred to any third party without written permission from Outside Technology.

It may not be distributed uncompiled in any form whatsoever.

You may not modify or create derivative works based on this software or accompanying materials and distribute them.

LIMITED WARRANTY. Outside Technology warrants that the software will perform substantially in accord with its defined functionality. This warranty is limited to 30 days following receipt. Outside Technology's entire liability shall be to repair or replace the Software which does not meet the limited warranty.

NO OTHER WARRANTIES: Outside Technology disclaims all other warranties, expressed or implied.

IN NO EVENT SHALL OUTSIDE TECHNOLOGY OR ITS OFFICERS OR REPRESENTATIVES BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING WITHOUT LIMITATION DAMAGES FOR LOSS OF BUSINESS) ARISING OUT OF THE USE, MISUSE OR INABILITY TO USE THIS SOFTWARE. OUTSIDE TECHNOLOGY'S ENTIRE LIABILITY SHALL BE LIMITED TO THE AMOUNT ACTUALLY PAID BY YOU FOR THIS SOFTWARE.

This Agreement is governed by the laws of the State of California.

DataLicense is a trademark of Outside Technology

4th Dimension, 4D Insider, 4D Compiler are registered trademarks of ACI/ACI US Inc.

Macintosh is a registered trademark of Apple Computer, Inc.

Windows is a registered trademark of Microsoft Corp.

Contents

Basics	5
What's Included?	5
Using the Demo Database	6
Converting a Demo Datafile	6
Installing DataLicense	7
Naming Conventions	7
All of the Steps, Briefly	7
Structure Setup	7
Getting the Parts In	8
Your Startup Routine	9
Two Methods You Must Modify	9
dID_INIT_DEV_SETTINGS	9
dID_SET_APPLICATION_CODES	10
Six Methods You May Modify	10
dID_INIT_STRUCTURE	10
dID_INIT_DEMO	10
dID_DataLicenseAccess	11
dID_EndDemo	11
dID_SEND_FILE	11
dID_ReceiveFile	11
Menu Items	11
Execute a Method	11
Odds & Ends	11
Security	11
Sounds	12
Style Sheets	12
Demo Datafiles, Trial Periods & Extensions	13
Demo Datafiles	13
Enabling Demo Mode	13
Limiting Functionalities	13
Converting a Demo Datafile	13
Trial Periods	14
The Datafile Create Date	14
Temporary Extensions	14
Modules	15
Module Setup	15
Licensing Modules	15
Enforcement	16
Compilation	16
Print Form Variables	16
The End-User Experience	17
First Log-On	17
Temporary Extensions	18
Serializing Datafiles	19
The Generator Application	20
Manual Generation	20
Automatic Generation	20
The Future	22
Modifying DataLicense	22
Possible Future Features	22
What are the chances of these actually happening?	22
Real-Life Concerns	23
Legal Issues	23
What if you Die?	23

Basics

The concept behind DataLicense is that your client or customer licenses your software to run ONE BUSINESS. What is actually licensed is the datafile, not the structure.

*DataLicense allows a datafile to be protected with a serial number that is unique to the Business Name of the client or customer; thus **preventing piracy of registered copies**.*

DataLicense also controls the number of users who may have concurrent access, and registers up to 32 optional modules.

With DataLicense installed, your 4th Dimension database operates in one of three modes: Demo, Trial, and Licensed. Demo mode is intended for finished programs offered for sale that ship with sample data and is probably not appropriate for custom solutions. Trial mode is intended for custom solutions, and is also useful in for-sale systems either as a step-up from demo mode or if sample data is not available.

This means:

- Every copy of your structure is a full working demo. If you include a demo datafile it will run indefinitely with whatever restrictions you place on demos. A trial datafile (one that is not tagged as a demo) will run unlicensed for a number of days after its creation that you specify. A demo datafile may be converted to a trial datafile by the end user.
- Only when the potential customer purchases a licence, or when the client pays you in full, do you give the serial number that allows permanent access to the datafile carrying the serialized business name. This allows you to easily provide trial periods of any duration.
- You may permit your clients and customers to freely distribute your application to others in their trade!
- DataLicense allows the developer to limit the number of users who may have concurrent access. You can eliminate the possibility that a customer will buy Server upgrades from someone else and not pay you for larger installations.
- Should a dispute arise between you and a client before you have been paid, at least the client will not have delinquent use of your work while the lawyers fight it out. It might even give you the extra leverage to bring a quick and satisfactory resolution.
- Using the optional postal code check will help prevent some situations where only one copy is bought, but is used by multiple locations of the same company.
- Registration for optional modules is included in the serial number. If you select modules when you generate the serial number the end user's application understands it transparently.

Besides the information in this manual, refer to the numerous comments in the methods themselves. To really understand the flow, place a TRACE in the method you want to understand and follow along under various conditions.

Please don't ship a database containing DataLicense until you feel you understand it.

What's Included?

DataLicense includes all methods and forms to make the system work in any 4D v6 structure, plus this User Guide. Methods and forms are provided in a 4D Insider library. If you do not own 4D Insider, you may copy methods and forms from the uncompiled demo. The uncompiled copy of the DataLicense Demo may be studied to understand the flow.

A separate 4D application that allows you to create serial numbers simply by entering the customer's business name is included.

The uncompiled copy of the DataLicense Demo contains several other interesting features, such as a version control system that guards against users inadvertently opening their datafiles with old copies of the structure, routines to automatically import data into empty tables such as Help, a text cleanup routine that strips garbage characters and leading and trailing spaces, plus replacements for some of 4D's standard dialogs.

Variables and methods are named for understandability, and the code is liberally commented and generally self-documenting.

Using the Demo Database

Experiment with the included demo database to gain an understanding of how DataLicense works at the end-user level before installing it in your real databases.

When a new datafile is created, it will run for 30 days before a serial number is requested. When you install DataLicense into your structure you may set this "free time" as you wish.

Use the Configure dialog to reset the number to -1 and reopen the database. You will be required to enter your serial number. (The request prompt shows the string that is encrypted.) Try to get in without a valid number. When you've given up trying to break in (it's just a matter of time), open the Generator application included with this demo and enter the string that appeared in the request exactly (it's case sensitive), click the Generate S/N button, copy the generated serial number, and use it to get into the datafile.

Now open the Return Address dialog. Notice that the datafile is licensed for one user. Use the Enter Serial button to add user licenses. A new serial number will need to be generated based on the license upgrade. If more users try to log on than are licensed, they will be informed as such and summarily terminated (so to speak).

A datafile may be tagged as a demo datafile, and that datafile will never ask for a serial number. (Be sure to limit the functionality of a demo datafile. If you do not want to limit functionality use DataLicense only for trial periods!) The configure dialog may be used to mark the current datafile as "Demo."

The Configure dialog may also be used to include the customer's postal code in the string that the serial number is built from. This is useful in controlling usage at a customer's multiple locations. When you install DataLicense in your applications you may substitute another field, such as address or phone, if you prefer. (*The Configure dialog is not intended to be installed in a shipping database!* Modify the dl.INIT_VARS method and run the dl.SET_DATAFILE_AS_DEMO method from the User environment to configure your program.)

Converting a Demo Datafile

If the end user clicks the Enter Serial Number button on the Return Address dialog when the datafile is a demo, DataLicense will convert the datafile.

You may also call the method dlM_KILLDEMO from a menu.

Installing DataLicense

Whichever method of installation you choose, **MAKE A BACKUP COPY OF THE STRUCTURE YOU ARE INSTALLING INTO!!!** *There is no way that we can anticipate the nature of your existing programming and it is possible that your database has characteristics that make it an inappropriate candidate for DataLicense.* Experiment on a copy just in case.

To avoid conflicts within your databases all methods, variables, tables, and forms have the "dl" prefix. You probably have methods that perform the same function as some of these, and you may replace them if you like for continuity.

Naming Conventions

All methods, tables, and forms that comprise the core DataLicense operations begin with the letters dl except for the methods COMPILER_DataLicense and COMPILER_dl_PARAMS. Interprocess variables begin with <>dl. Arrays begin with <>a_dl. The few process variables are extremely localized and begin with dl except for button variables.

There are 7 methods that begin with dld_. They are safe for you to modify: DataLicense upgrades will not overwrite these methods. You must modify dld_INIT_DEV_SETTINGS and dld_SET_APPLICATION_CODES. You may modify dld_DataLicenseAccess, dld_EndDemo, dld_InitDemo, dld_INIT_STRUCTURE, and dld_SEND_FILE.

Two methods begin with dlm and they are commands which should be called from your programs' menu. dlm_EDIT_ADDRESS is mandatory and dlm_KILLDEMO is optional.

Two methods begin with dlu and are provided for you to execute from 4D's User environment. dlu_SEND_APPLICATION_CODES is mandatory and dlu_SET_DATAFILE_AS_DEMO is optional.

All of the Steps, Briefly

First, a quick list of what must be done:

- Move everything into your database, preferably using 4D Insider™.
- Add the method dl_STARTUP to your startup method.
- Edit the methods dld_INIT_DEV_SETTINGS and dld_SET_APPLICATION_CODES, and examine the other 4 dld methods to determine if you want to modify them.
- Add a menu item for dlm_EDIT_ADDRESS and, optionally, dlm_KILLDEMO.
- Execute the method dlu_SEND_APPLICATION_CODES from the User environment and move the *.dlm file to the Applications folder in the Generator folder.

And now, the details...

Structure Setup

If you install the DataLicense group from the Insider library, the table dlConstants is created. It contains fields to store information about the end-user's business and some data that DataLicense needs. *This must be a one-record table.*

If you do not own 4D Insider, create the table defined below in your database, or refer to other fields that serve the same function by assigning them to the appropriate pointer variables in the method dld_INIT_STRUCTURE, and the fields on the form dlInput.

The table [dlConstants] contains the fields:

[dlConstants]Name	A60
[dlConstants]Address	A40
[dlConstants]Address2	A40

[dlConstants]City	A20	
[dlConstants]State	A2	
[dlConstants]PostalCode	A10	
[dlConstants]Phone	A14	
[dlConstants]Fax	A14	
[dlConstants]Logo	P	
[dlConstants]StartDate	D	INVISIBLE
[dlConstants]Seats	I	INVISIBLE
[dlConstants]SerialNo	A20	INVISIBLE
[dlConstants]Modules	A8	INVISIBLE

After these, there are 6 Reserved fields contained in the table installed with Insider. If you want to add fields to this table please do not use the reserved fields. You may add as many as you like after them.

The actual names and alpha sizes do not need to match exactly, but they should be functionally the same. The size of State, PostalCode, and Phone fields may be changed for localization purposes. None of the fields need indexes or other attributes to be set except for the last four which should be invisible.

The field [dlConstants]Modules must be increased in length if you offer more than 8 optional modules. The absolute maximum is 32. If you change the length of this field, any clients running serialized databases will have to be issued a new number.

If your database already has a one-record table that stores the user's name and address info, just add the last 4 fields (StartDate, Seats, SerialNo, and Modules) to it. If you choose to do this and you install using the Insider library, select everything *except* the table and the group and control-drag to bring up a dialog that asks what you want to do about the missing table. Select "Use another object" and then choose the table you want to use.

The only places that the fields are directly referenced are the method `did_INIT_STRUCTURE` and the form `dlInput`. So if you use your own table you must change the references in these areas.

No more than one record can exist in this table. Do not allow users to add records to this table. DataLicense checks for multiple records at startup and if more than one is found will try to delete only inconsequential records, but if that fails will delete them all.

Getting the Parts In

If you have 4D Insider, be sure it's allocated at least 16 MB (on Mac).

Open Moving Preferences from Insider's File menu. These are the recommended settings:

- * A form with the same name already exists.
-> Replace
- * A method with the same name already exists.
-> Replace
- * An identical menu bar already exists.
-> Do not copy
- * This menu bar # already exists.
-> Do not copy
- * An identical menu item already exists.
-> Do not copy
- * A format/filter with the same name already exists.
-> Replace
- * A picture with the same number already exists.
-> Do not copy
- * A style sheet with the same name already exists.
-> Do not copy
- * A similar table does not exist.
-> Copy the object (Unless you want to reference an existing table.)

- * There is no table in the destination structure.
-> Copy the object
- * A similar field does not exist.
-> Add a field to the table
- * There is no field in the destination table.
-> Add a field to the table
- * A form with the same name does not exist.
-> Copy the object
- * There is no form for this table.
-> Copy the object
- * A similar method does not exist in the destination structure.
-> Copy the object
- * An identical menu bar does not exist.
-> Ignore
- * There is no menu bar .
-> Ignore
- * This filter/format does not exist.
-> Copy the object
- * A style sheet with the same name does not exist.
-> Copy the object

Open your database with Insider 6.0.6 or later. Open the DataLicense library. Select "Groups" from the popup at the middle small pane. Drag the DataLicence group icon to the middle small pane of your structure.

If you don't own 4D Insider, consider buying it. Everyone that uses it *knows* how indispensable it is.

If you don't have it and are not going to buy it, you may copy and paste from the uncompiled demo database. You need to copy all methods and forms beginning with dl plus the methods COMPILER_DataLicense and COMPILER_dl_PARAMS.

Your Startup Routine

Open your database and open (or create) the Startup method.

The method dl_STARTUP must be incorporated into your Startup method, preferably at the beginning.

Two Methods You Must Modify

dlID_INIT_DEV_SETTINGS

dlID_INIT_DEV_SETTINGS allows you to set values of several variables that control behavior of the serial number mechanism. The method comments explain the use of each.

◊**dlAppName**: The name of this program. Enter manually or use
◊ dlAppName:=*dl_ShortName* (**Structure file**)

It is preferable to hard-code the app name to avoid confusion if an end-user gets creative.

◊**dlDeveloperContact**: A tag line containing your phone number or email, etc. instructing the user how to get help (purchase a copy!) It appears in dialogs after an unsuccessful attempt to gain access.

◊**dlDeveloperName** Your business name. This must match the Business Name in your copy of Generator.

◊**dlMail_FaxInfo** This address & fax number appears on the "dlPrintRegRequest" form

◊**dlLeeway**: The number of days after StartDate (the first use of your program by a new client or prospect) before a serial

number is required in a trial or new datafile. The free *full-use* period.

◊**dlExtendDaysDefault**: Default days+current date for temp extensions. This is a default, you may extend to any date in any individual case.

◊**dlMaxTries**: Maximum number of times you'll allow a user to guess before kickout (max incorrect tries). This is important to foil automated sequential generators.

◊**dlUsePostalCode**: Optional. Set this false to NOT include the users postal code. Use it to limit licensed use to one location

◊**dl_SN_Universal**: Concoct your own gibberish. Have your lawyer send this code to your clients when you die or win the lottery. Anybody entering this code gets permanent unrestricted access.

◊**dl_DeleteDemoData**: This causes demo data to be deleted when the user decides to create a real datafile and start the full-use trial period

◊**dlUpgradeExtensionsOK**: if true, allows temporary extend-to dates to apply to licensed datafiles. This will allow unlimited connections through the extend date, Possibly useful if a client has committed to upgrade but has not yet fully paid for it. You want to think about this...

◊**dlModuleDirectSelect**: When the end user is entering a serial number, a selection array of available modules is presented if ◊dlModuleDirectSelect is true. If no list is presented DataLicense will test the S/N entry to determine the modules to register. If ◊dlModuleDirectSelect is false (the default behavior), the user may still bring up the list if the command (Mac) or Control (Win) key is down when clicking the Enter Serial Number button.

dlID_SET_APPLICATION_CODES

Assign a random 6 or 7 digit longint to ◊dlApplicationCode. This number is passed to the Generator application and is also used in validating an entered serial number. If you change this number, you must execute dlu_SEND_APPLICATION_CODE again (see below), and any clients running serialized databases will have to be issued a new number. You *must* use different numbers in each of your applications.

Add an element to the ◊a_dlModuleName array for each available optional module. Please read the Modules chapter for more information.

Six Methods You May Modify

dlID_INIT_STRUCTURE

Establishes pointers to fields DataLicense needs. The rest of the code references these pointers, so if you choose to use an existing (one-record!) table instead, this (and the dlInput form) are the only connections you need to make.

The table name and field names do not have to be the same, but they should represent the same data or you will need to modify a number of things, such as dl_SET_RETURN and the dlInput form. Be sure all fields are in the same *one-record* table and that the pointer symbol (->) is used.

dlID_INIT_DEMO

If you are not offering a demo mode you do not need to edit this method.

This method runs at every startup if the datafile is a demo. Add any code that sets demo behavior. See the next chapter for suggestions on limiting the functionality of a demo database.

If you call dlM_KILLDEMO from a menu, edit the alert to let the users know where to find it. If you do not offer that menu item, remove the alert line.

dlID_DataLicenseAccess

Allows you to set access to dlTable-> for the current user. Possible values are "ReadWrite", "ReadOnly", or no access. If you do not modify this method the default is ReadWrite for all users.

dlID_EndDemo

If you are not offering a demo mode you do not need to edit this method.

This method runs when the end user chooses to convert a demo datafile to a real one. There is code included that deletes demo data if the `dl_DeleteDemoData` variable is set to true in `dID_INIT_DEV_SETTINGS`. Any other code you may want to execute when a datafile is converted should be added.

dID_SEND_FILE

Out of the box, DataLicense allows the end user to click a button ("Request Serial Number" on the Return Address (dlInput) form) that creates a text file on disk that Generator can read and use to generate a serial number and create a registration record. As shipped, `dID_SEND_FILE` creates a folder on the desktop (Mac) or C drive (Windows) and saves the file to that folder.

You may modify `dID_SEND_FILE` to operate at another level of automation based on your communications tools. Examples might include sending the file as an email attachment or to an ftp directory.

dID_ReceiveFile

The counterpart to `dID_SEND_FILE`, this method controls where DataLicense looks for the return file you send from Generator.

Menu Items

Add an item "Edit Return Address..." or "Edit Company Information..." to the File menu at your program's home screen. Assign the method `dIM_EDIT_ADDRESS` to it. It is recommended that access to this item be restricted to administrator-level users.

If you are using demo mode and would like to offer users a menu command to convert a demo datafile to a trial mode, the method `dIM_KILLDEMO` is provided. It is recommended that access to this item be restricted to administrator-level users.

There are many ways to manage menus in 4th Dimension and DataLicense cannot presume to know how you have yours set up. It is recommended for interface reasons that you hide the command that calls `dIM_KILLDEMO` if the datafile is not a demo.

Execute a Method

From 4D's User environment, execute the method `dlu_SEND_APPLICATION_CODE`. The values you set in `dl.SET_APPLICATION_CODES` are packed into a BLOB and saved as the disk file `<AppName>.dlm`.

Move the `.dlm` file into the Applications folder in your Generator folder.

Odds & Ends

Security

It is recommended that the menu items that call `dIM_EDIT_ADDRESS` and `dIM_KILLDEMO` be restricted to Administrator access. Malicious or careless users could easily cause problems otherwise.

All DataLicense methods are set to invisible except `dlu_SEND_APPLICATION_CODE` and `dlu_SET_DATAFILE_AS_DEMO`, since they are intended to be run by the developer from the User environment. We do not want end users to have any access to these methods, so set them to invisible in method properties before shipping your database.

Sounds

Insider cannot move sound resources into your databases, so if you want DataLicense to play its cute little sound effects open the file DataLicense Sounds with ResEdit or Resourcerer and copy and paste all four into your structure file. Don't change the sound names.

Obviously, you all-Windows developers are gonna have to buy a Mac. ;-)

Style Sheets

Four style sheets are installed with DataLicense: dlSmall, dlLarge, dlPrintSmall, and dlPrintLarge. Every DataLicense object with font properties is assigned to one of these style sheets.

Change the style sheet definitions to make the DataLicense forms consistent with the rest of your application, instead of changing the fonts of the objects themselves.

Demo Datafiles, Trial Periods & Extensions

Demo Datafiles

A datafile tagged as a demo will run forever, with no serial number enforcement!

Test the method `dl_Is_Demo` anywhere in your code for the status of the datafile.

```
If (dl_Is_Demo)
    ALERT ("This is not available in the demo version.")
Else
    `create a record, access a feature, etc...
End if
```

Enabling Demo Mode

Create a datafile with sample data and execute the method `dlu_SET_DATAFILE_AS_DEMO` from the User environment. From the runtime ("custom menus") environment open the Return Address dialog and enter bogus company info for the sample.

Limiting Functionalities

If you tag a datafile as being a demo DataLicense will never check for a serial number, so you must ensure that some important functionalities are limited.

The method `dID_INIT_DEMO` is provided for you to use to establish demo restrictions.

Some common ways of limiting a demo are to set a string variable that appears on print forms to the word "DEMO," limit the number of records that can be added to certain tables, or other limitation appropriate to the application.

If you choose to limit the number of records, it may be a good idea to enforce the limit on only one or two key tables based on your program, Invoices or Appointments for example.

A very effective technique is to keep all date-related info current. Check the Appointments, Invoices, Events, etc. tables for dates less than **Current Date** minus x days. If any are found, adjust the dates of all records so that they are future.

The keep-dates-current idea is powerfully effective for three reasons. First, it will *make sure a prospect examining a demo will see **current** data regardless of when they use the demo.* Second, it means that *all functionalities can be enabled.* And third, it will (of course) *render the datafile useless for any purpose except demonstrating the software.* Just be sure to throw up an alert to let the prospect know what is happening so they don't think date shifts are caused by a bug.

Converting a Demo Datafile

DataLicense provides a mechanism to convert a demo datafile to a real one.

It is invoked when the user clicks the Enter Serial Number button on the Return Address form, after asking if the user wants to convert the datafile.

The method `dIM_KILLDEMO` is also provided for you to attach to a menu item.

Under both, the method `dID_EndDemo` is called. You may modify `dID_EndDemo` to control what happens when the conversion takes place.

If the `ddl_DeleteDemoData` variable is set to true (`dID_INIT_DEV_SETTINGS`) all demo data will be deleted during the conversion. Do not use this option unless you are providing a datafile with demo data.

If the `ddlLeeway` variable is greater than zero, the trial period will begin now (unless, of course, the user enters a serial number).

Trial Periods

A datafile which is neither a demo nor serialized is considered a trial datafile and will be able to be opened only within the date range bounded by the datafile create date and the create date plus the `ddlLeeway` variable, set in `dID_INIT_DEV_SETTINGS`. If `ddlLeeway` is greater than zero every new datafile will run in trial mode.

DataLicense believes that trial mode functionality should be unrestricted, the only constraint being the date range.

The Datafile Create Date

DataLicense sets the start date of the datafile when the datafile is created, or when a demo datafile is converted.

To prevent a wily user from having the computer's system clock set to the year 2525 when creating a datafile, thereby gaining 500 or so free years of use, DataLicense asks for a confirmation of the current date when making this setting. On subsequent startups, the program will not run if the current date is earlier than the create date.

Trial mode continues until `CreateDate` plus `ddlLeeway` is reached, at which time a serial number is required for entry at startup.

Temporary Extensions

DataLicense allows temporary extensions of the trial period. This is useful in cases of ongoing custom development, check-is-in-the-mail scenarios,

After cancelling the dialog that requests entry of a serial number, the user is asked if they have an extension. They may enter the extended-to date, then the code. The date is formatted for decryption as Month Day Year (with no separators) regardless of the system date format so that there is no confusion when you create the key. For example, April 14, 1999 will always be "4141999".

Modules

Module Setup

Set up modules in `dID_SET_APPLICATION_CODES`.

Size the `∅a_dIModuleName` array to the number of modules offered in this database and assign the name of the module to each element.

```
ARRAY TEXT(∅a_dIModuleName;5)
∅a_dIModuleName{1}:="Web Publishing"
∅a_dIModuleName{2}:="Email"
∅a_dIModuleName{3}:="Inventory"
∅a_dIModuleName{4}:="Complete Accounting"
∅a_dIModuleName{5}:="Mail Merge"
```

If you need to register more than 8 optional modules, the field `[dIConstants]Modules` must be increased in length, up to a maximum of 32. If you change the length of this field the encryption scheme changes, so you must execute `dIu_SEND_APPLICATION_CODE` again (see below), and any clients running serialized databases will have to be issued a new number.

Adding elements to the module array without changing the length of the `Modules` field does not require the codes to be sent or existing datafiles to be re-serialized.

The method `dI_ModuleOK` has been provided so you may test the status of a module anywhere in your code. The string passed to `dI_ModuleOK` must exist as an element in the `∅a_dIModuleName` array.

```
If (dI_ModuleOK ("Inventory"))
  `do inventory stuff
End if
```

Licensing Modules

If the user clicks the Request Serial Number button on the Return Address dialog, they may select the modules they want. Generator will receive that information when it reads the disk file.

If you manually enter the user's information into Generator you must select the purchased modules before generating the serial number.

When the serial number is either entered by the user or the registration file read from disk, `DataLicense` will interpret the modules to enable. Alternately, the user may Control-click (Windows) or Command-click (Mac) the Enter Serial Number button to allow manual module selection at the time of entry. Or, you may specify that serial number entry always allows direct module selection by setting the `∅dIModuleDirectSelect` variable true in `dID_INIT_DEV_SETTINGS`.

Enforcement

Compilation

Does this need to be said? To be secure, *your database must be compiled*.

Print Form Variables

A DataLicense serial number is valid only for the business name entered in the Return Address dialog.

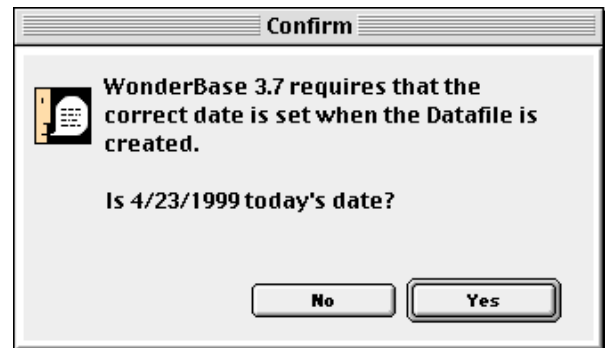
To make that serial number useless to any business except the registered user, the values entered in the Return Address dialog should appear on important print forms such as invoices and merge letters.

DataLicense generates the variables `®dlBusinessName`, `®dlReturnAddress`, and `®dlFooter` at startup. They're built in the method `dl_RETURN_ADDR` and may be found on the sample print form "dlPrintTemplate". To complete that DataLicense security circle, please use them.

The End-User Experience

First Log-On

When a new datafile is created, or the user has executed the Convert Demo menu item, or has opened their datafile for the first time after DataLicense installation, they must confirm the current date.

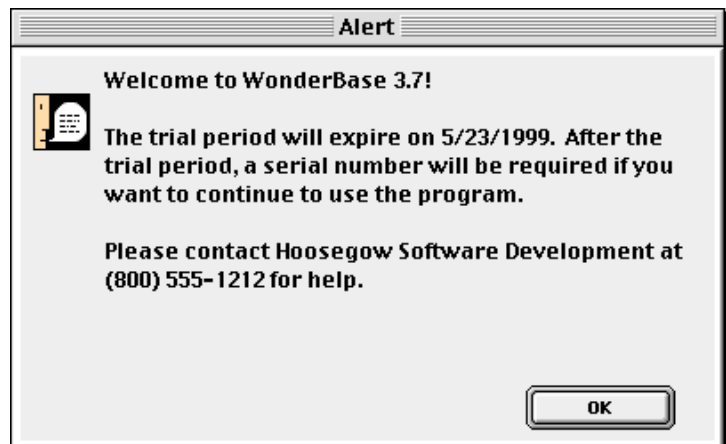


The Return Address entry form opens. The Business Name and Postal Code are required.

A form titled "Return Address". It contains several input fields: "Business Name" (filled with "BozoDyne Industries, Inc."), "Address" (filled with "345 Twertvale Blvd."), "Address2" (filled with "Suite 337"), "City.State.PostCode" (split into "Vapid City", "NB", and "55016"), "Country" (empty), "Phone" (filled with "(613) 555-1212"), "Fax" (filled with "(613) 555-2211"), "eMail" (filled with "info@bozo.com"), and "Web Address" (filled with "www.bozo.com"). There is a "Logo" placeholder on the right. At the bottom left, it says "Unregistered" in purple. At the bottom, there are four buttons: "Enter Serial Number", "Request Serial Number", "Preview", and "Save".

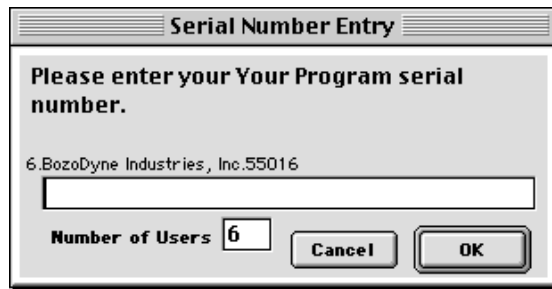
If the Return Address form is saved without a serial number entry, and the \diamond dlLeeway variable is greater than zero, the user receives a welcome message and information on the expiration of the trial period.

DataLicense will not pester the users with further alerts until the trial period has expired.



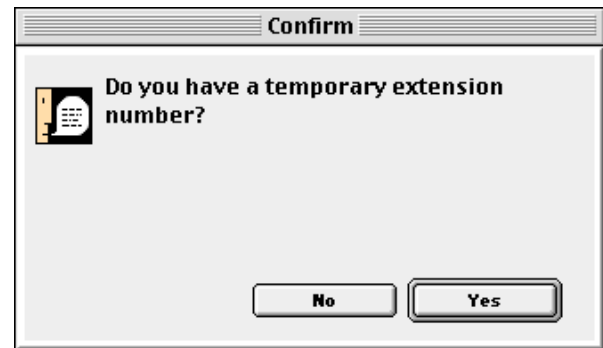
Temporary Extensions

If the normal grace period is not enough, you may extend the trial.

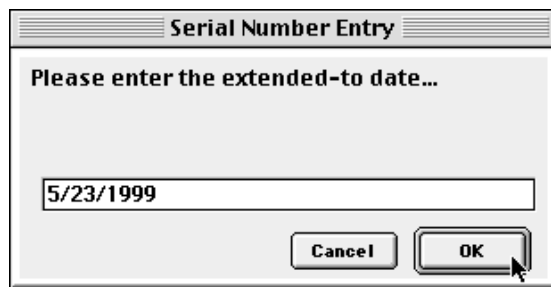


The user clicks the Enter Serial Number button on the Return Address dialog or receives the serial number request at startup after the trial period has ended.

Not having a number, the user clicks Cancel.



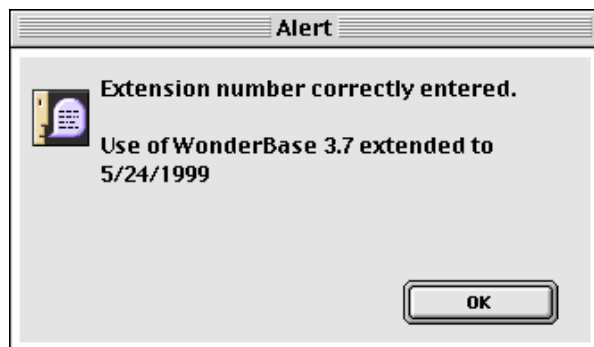
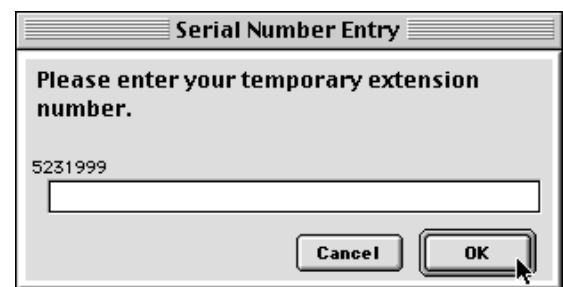
They are asked if they have a temporary extension.



We need to know the extended-to date. The `dlExtendDaysDefault` variable (set in `dlID_INIT_DEV_SETTINGS`) plus the current date sets the default, but you may extend them to any date you like.

You will use Generator to create the code that will continue the trial period.

The date string that you must use to generate the code is always returned in Month - Day - Year order to prevent confusion. No separators or spaces are used. The actual number that you must use to generate the code is shown directly above the prompt entry area.

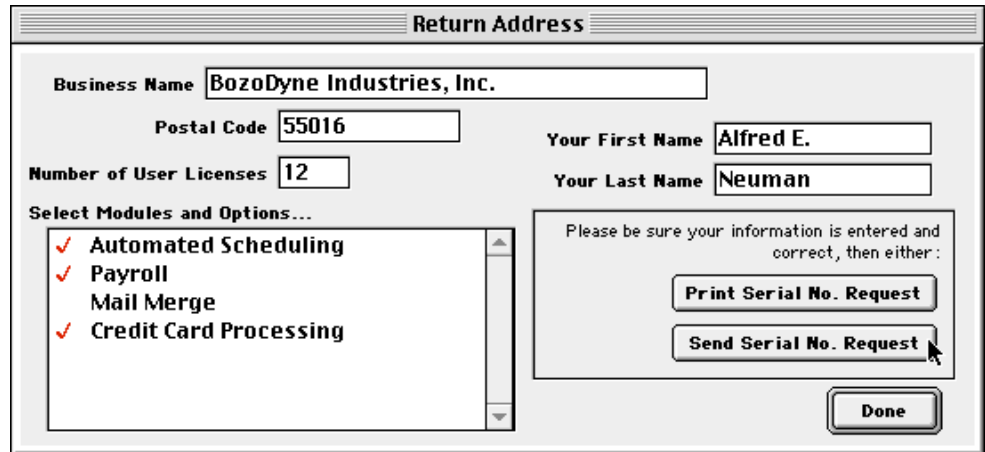


Serializing Datafiles

You may, of course, manually enter the user's request string in Generator, create a serial number, and communicate it to the client. Likely, you'll do this over the phone occasionally for users whose trial period has ended and need to get into the program.

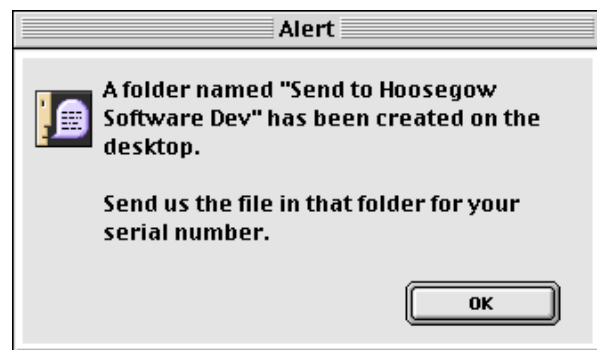
DataLicense does, however, provide an automated mechanism that you may prefer to use.

The user clicks the Request Serial Number button on page 1 of the Return Address form, enters the number of user licenses they want to purchase, selects the modules they want, and clicks the Send Serial Number button.

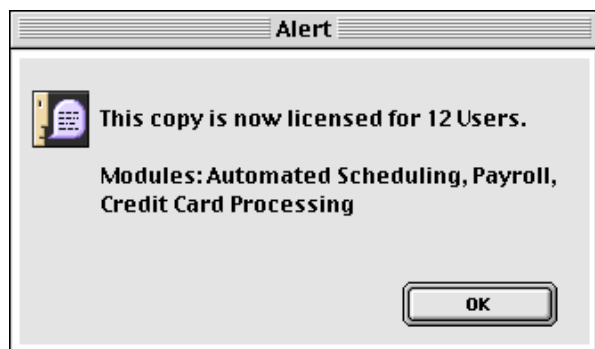
A screenshot of a 'Return Address' form. It contains fields for 'Business Name' (BozoDyne Industries, Inc.), 'Postal Code' (55016), 'Number of User Licenses' (12), 'Your First Name' (Alfred E.), and 'Your Last Name' (Neuman). There is a section 'Select Modules and Options...' with a list box containing 'Automated Scheduling', 'Payroll', 'Mail Merge', and 'Credit Card Processing', all of which are checked with red checkmarks. To the right of the list box is a text box with the instruction 'Please be sure your information is entered and correct, then either :'. Below this text box are two buttons: 'Print Serial No. Request' and 'Send Serial No. Request'. At the bottom right of the form is a 'Done' button.

DataLicense creates a folder on the desktop (Macintosh) or C drive (Windows) and saves a registration text file into the folder. (You may modify the method dID_SEND_FILE to change where the file is saved or how the file is sent to you.)

All you need to do is open this file with Generator (See the next chapter), and click the Send S/N to Disk button, and get the serial number file to the user.



Any time DataLicense needs a serial number, which can be at the first startup after expiration of the trial period or whenever the user clicks the Enter Serial Number button, DataLicense checks for a Generator export file before asking the user to enter the number. The structure folder (if single-user) or application folder (if 4D Client) is checked first, then the desktop of the startup volume, then root level of the startup volume. Any of those locations is OK.



If the file is found and validated, registration is automatic and the export file is moved into a folder named SerialArchive (automatically created by DataLicense) in the structure or application folder. The file is moved to prevent reading of old files during subsequent upgrade serializations.

The Generator Application

For every 4D structure you want to generate serial numbers for, there must be a file in Generator's Applications folder. Please read about the `dlID_SET_APPLICATION_CODES` and `dl_SEND_APPLICATION_CODES` methods in the Installation chapter.

Generator is a compiled database and is itself protected by DataLicense. The return address info you enter into Generator should be your own.

Manual Generation

To generate a serial number using manual entry, select the application the client is using, select any optional modules purchased, and type in the string exactly as it appears in your client's request prompt. This is case-sensitive. Click the Generate S/N button (Command or Control-G).

The users' string must be fully entered and the modules must be selected before the serial number is generated.

Or, use the Build String button for a series of prompts allowing successive entry of number of users, business name, and postal code. If the Build String button is used, the same auto capitalization function used in DataLicense's Return Address dialog is used.

When you give the number to the client, remember to distinguish between zero and the letter O, and 1 and the letter I. The serial number format is always LL-NNNN-NLNNN.

If you are generating a code for an extension date, the separators must be stripped and the order month day year is always used. If the extension date is 9/18/99, you decrypt 9181999.

To enter a registration record in the Client database, click the Create or Update Record button and enter manually.

Automatic Generation

Obtain the file created by the user (see the section Serializing Datafiles in the End User Experience chapter). Use the Read Disk File button to open it.

Pertinent info from the user's installation is acquired, and a serial number is generated.

The screenshot shows the 'Generator' application window. At the top, the 'Application' dropdown is set to 'WonderBase 3.7'. Below this, the 'Modules & Options' section lists three checked items: 'Automated Scheduling', 'Payroll', and 'Credit Card Processing'. To the right, a text area contains contact information for Alfred E. Neuman at BozoDyne Industries, Inc., including address, phone, fax, email, and website. Below this, it shows 'MacOS version 8.5.1' and '4th Dimension 6.0.6 F1'. A 'FileSaved' button is in the top right. In the center, the 'User's Request String' field contains '12.BozoDyne Industries, Inc.55016'. A yellow warning box below it states 'The User's Request String is CASE SENSITIVE!'. At the bottom, the 'Serial Number' field displays 'LC-6618-0F620'. Below the serial number, it shows the format 'LL-NNNN-NLNNN' and a note: 'The first 2 chars and the 2nd char of the 3rd block are always letters. All others are numbers.' Buttons for 'Print Registration', 'Send S/N to Disk', and 'Generate S/N' are present. A 'Create or Update Record' button is also visible. At the bottom right are 'Clear Form' and 'Done' buttons.

Click the Send S/N to Disk button to create a file in Generator's Serial Numbers folder. Send this file to the user. Its presence on disk will automatically serialize their datafile.

Click the Create or Update Record button to enter the data received in the import file in the Client Registration database.

Clients: 8 of 8

Clients Existing Record: 1 of 8

Name

Address

Address2

City

State

Zip

Country

Phone

Fax

eMail

URL

Notes

Registrations	Seats	RegistrationDate
WheatPro98	6	9/17/1998
WonderBase 3.7	12	4/23/1999

Contacts	
Alfred E.	Neuman

The Future

Modifying DataLicense

Please keep notes on any changes you may make to DataLicense code someplace other than as comments to the code itself. If you upgrade to a future version, any comments you add to DataLicense core code (except, of course, the dID_ methods) will be replaced along with the code itself.

Possible Future Features

- DataLicense may include some routines to help enforce limits on demo datafiles.
 - DataLicense may automate the extension process.
 - Allow the Send Serial Number Request function to understand pricing details of single user, server seat, modules, and version upgrade purchases. This will allow the user to price their package and include a credit card number in the export file. Generator could contain hooks to Tellan Software's MacAuthorize/PC Authorize.
- Generator's mechanism could then be fully automated to monitor a Watch folder, read any new files, process the credit card information, create/update registration records, and send out serialization files.
- Generator may evolve towards a full-featured consultants database incorporating time & billing, estimating, support & bug logs, inquiry management, etc.

What are the chances of these actually happening?

It depends on sales of this version.

Hundreds of hours went into producing DataLicense. Please do not give or sell the uncompiled code or Insider library to anyone. Thanks!

Real-Life Concerns

DataLicense can have serious consequences if you do not think about what you are doing. You must be aware of some things.

Legal Issues

To avoid unpleasant surprises, misunderstandings, and potential lawsuits, it is essential that your clients and customers understand from the beginning that they are buying a license to use your software to run one business and that the serial number is based on their business name as entered in the program.

They must also be informed that the program will run for x days before a serial number is requested, and that it is this fact that enables you to offer a money-back guarantee or trial period (on vertical market applications) or credit (on custom work.) Undisclosed time bombs can have serious consequences. If they don't pay we can stop them from using the program, but legally they OWN their own data.

If the trial period ends and you haven't been paid, causing you to refuse to provide a client or customer with a serial number, DataLicense will offer to export all data (except subtables, pictures, and BLOBs) to disk before quitting. You may control which tables are included in this export by modifying the `dl_EXPORT_ALL_TABLES` method.

What if you Die?

Sooner or later it *could* happen, right? Don't laugh.

You've sold some programs and given 60-day money-back trials. The customers are running their businesses with your software and they love it. The trial period expires, they get the request for a serial number and phone you for it.

Other clients try to contact you for a Server upgrade.

But wait! You're dead.

If you're a one-man-shop and have not bequeathed your applications and clients to another developer, assign an impenetrable value to the `dl_SN_Universal` variable (`dl.INIT_VARS` method) and print the value in a note for each of your clients. Put them in addressed envelopes and leave them with a trusted associate, friend, attorney, or family member and instruct them to mail the envelopes if you die or become incapacitated. If you win the lottery you can mail 'em yourself.

This way, customers who have paid in full long ago will be able to change their business names or postal codes, use all modules, and freely upgrade their Server license.