

da Vinci

The Data Architect for Palm OS
Version 1.0

User's Guide

Designed and distributed by ETS, Inc.
2531B State St.
Carlsbad, CA. 92008
<http://www.exts.com>



END USER LICENSE AGREEMENT

PLEASE READ THIS LICENSE AGREEMENT CAREFULLY BEFORE USING THIS SOFTWARE. BY USING THIS SOFTWARE, YOU AGREE TO BE BOUND BY THE TERMS HEREOF. IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, PROMPTLY RETURN THE ENTIRE SOFTWARE PACKAGE WITH ALL OF ITS CONTENTS TO THE PLACE OF PURCHASE FOR A FULL REFUND.

1. License and Warranty: The software which accompanies this license (the "Software") and the related documentation is licensed to you by ETS, Inc. ("ETS"). ETS, and or ETS Licensor(s), and suppliers retain title to the Software and related documentation, and is protected by United States copyright laws, by laws of other nations and international treaties. This license provides you with the following rights:

You may:

- (I) use one copy of the Software on a single computer;
- (II) make one copy of the Software for backup purposes, or copy the software onto the storage media of your computer and retain the original for backup purposes;
- (III) use the Software on a network, provided that you have a licensed copy of the Software for each computer that accesses the Software over that network;
- (IV) transfer the Software on a permanent basis to another person or entity, provided that you retain no copies of the Software and the transferee agrees to the terms of this license agreement;
- (V) the right to use the Software for the sole and limited purpose of generating source code for the development of custom and commercial software applications, without any royalty or compensation requirements to ETS.

You may not:

- (I) copy the documentation which accompanies the Software;
- (II) sublicense, rent or lease any portion of the Software;
- (III) reverse engineer, decompile, disassemble, modify, translate, make any attempt to discover the source code of the Software, or create derivative works from the Software.

2. This License is effective until terminated: You may terminate this License at any time by destroying the Software and related documentation. This License will be considered terminated without notice from ETS if you fail to comply with any provision of this License. Upon termination you must destroy the Software and related documentation.

3. Maintenance and Upgrades: This License is for the Software only, and except where expressly provided, ETS hereby expressly disclaims any obligation to provide maintenance, patches, or fixes to the Software. ETS may, at ETS's discretion, prepare and distribute modifications, enhancements, new releases, and updates of Software, under terms and conditions as ETS may determine, acting in its sole discretion.

4. ETS Obligations: ETS reserves the right to transfer its rights and delegate its duties, liabilities or obligations to a third party without release of its rights hereunder or prior notice to you, and such assignment and delegation shall be deemed a release by you of ETS of all obligations hereunder.

5. ETS Will: Replace defective disks or manuals within 90 days of the date of purchase, at no charge to the end user.

6. You may NOT disclose or share your software registration key with any individual or group, and you agree that such restriction on disclosure shall remain in effect for a period of five years from the effective date hereof.

7. Disclaimer of Warranty on Software. You expressly acknowledge and agree that use of the Software is at your sole risk. The Software and related media materials are provided "AS IS" and without warranty, of any kind and ETS and ETS's Licensor(s) expressly disclaim all warranties, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. ETS does not warrant that the functionality contained in the Software will meet your requirements, or that the operation of the Software will be error-free, or that defects in the Software will be corrected. ETS does not warrant or make any representations regarding the use or the results of the use of the Software or related media materials in terms of their accuracy, reliability, or otherwise. No information or advice, whether oral or written, given by ETS or ETS's Licensor(s) shall create a warranty, or in any way increase the scope of this warranty. Should the Software prove defective, you (and not ETS or ETS's Licensor(s)) assume the entire cost of all necessary servicing, repair or correction. Some states do not allow the limitation or exclusion of liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. In no case shall ETS's liability exceed the purchase price for the Software. The disclaimers and limitations set forth above will apply, regardless of whether or not you accept the Software.

8. Risk: You acknowledge and agree that this Agreement allocates risk between you and ETS as authorized by the Uniform Commercial Code and other applicable law, and that the pricing of ETS's products reflects this allocation of risk and the limitations of liability contained in this Agreement. If any remedy hereunder is determined to have failed of its essential purpose, all limitations of liability and exclusions of damages as set forth in this Agreement will remain in effect.

9. Limitation of Liability: Since software design and development is inherently complex and may not be free from errors, you are advised to verify the work produced by the Software. Under no circumstances, including negligence, shall ETS be liable for any incidental, special, or consequential damages that result from the use or inability to use the Software or related media materials, even if ETS or ETS's Licensor(s) has been advised of the possibility of such damages. Some jurisdictions do not allow the limitation or exclusion of liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. In no event shall ETS's total liability to you for all damages, losses, and causes of action, whether in contract, tort (including negligence) or otherwise, exceed the amount paid by you for the Software and related media materials.

10. Support: ETS may, at its option, provide support services at its standard fees for such services. Such support services will be governed by the limitations of liability under this Agreement.

11. Controlling Law and Severability: This Agreement will be governed by the laws of the State of California. This Agreement may only be modified by a license addendum which accompanies this license, or by a written document which has been signed by both you and ETS. Should you have any questions concerning this Agreement, or if you desire to contact ETS for any reason, please write ETS, Inc. Customer Sales and Service, 2531B State St., Carlsbad, CA. 92008

Copyright © 2002 ETS, Inc. and its licensors. All rights reserved. No part of this documentation may be reproduced or transmitted in any form without express written permission.

Documentation Disclaimer:

The author and publisher of this documentation have used their best efforts in preparing this material. ETS, Inc. makes no representation or warranties with respect to the accuracy or completeness of the contents of this documentation, and specifically disclaim any implied warranties of merchantability or fitness for any particular purpose, and shall in no event be liable for any loss of profit or any other commercial damage, including but not limited to special, incidental, consequential or other damages.

Printed in the United States of America
First Printing, June, 2002
Distributed in the United States by **ETS, Inc.**

Copyrights, Trademarks, and References

da Vinci, the Data Architect for Palm OS, is copyright © 2002, ETS, Inc.

Metrowerks and CodeWarrior are trademarks of Metrowerks Corp in the US and other countries.

Microsoft Windows NT, ME, 2000, XP, are registered trademarks of Microsoft Corporation in the US and/or other countries.

Palm OS is a registered trademark of Palm, Inc.

All other trade names and trademarks are the property of their respective owners.

Contents

END USER LICENSE AGREEMENT	2
Copyrights, Trademarks, and References	4
User's Guide Overview	8
1 – Welcome to da Vinci, the Data Architect	9
What is da Vinci?	9
The Birth of da Vinci	9
Why Use da Vinci?	9
Why Use da Vinci Instead of A Database Engine?	10
How Efficient Is the Code from da Vinci?	10
Will da Vinci Write a Complete Application?	10
2 – Getting Started with da Vinci, The Data Architect	11
Minimum PC Requirements	11
Recommended User Prerequisites	11
Installing da Vinci	11
Electronic Distribution Version	11
CD ROM Distribution Version	11
3 – Palm OS Databases	12
4 – da Vinci Tutorial	13
Quick Start	13
5 – The da Vinci User Interface	27
da Vinci Toolbar	27
da Vinci Menus	28
6 – Creating and Managing a Project	29
Creating a Project	29
7 – Creating and Managing Databases	30
Functions	30
New Database	30
Remove Database	30
Preference Database	30
8 – Creating and Managing Fields	32
Functions	32
Add a Field	32
Remove a Field	33
9 – Project Properties	34
Main Properties Fields and Functions	34
Project Name	34
Version	34
CreatorID	34
Project Save Path	34
Palm OS Header File	34
Include Function Comments	35
Include Line Comments	35
Show Code Generation Progress	35
DB Open/Close Properties	35
Database Open/Close Support	35
Open Database By	35
Create New Databases	36
All Databases	36
Select Databases...	36

Available DB's	36
Selected DB's	36
Remove Database	36
DB Sorting Preferences	36
Include Database Sorting Support	37
Available DB's	37
Sort This Database	37
Sort Field(s)	37
Global Find Preferences	37
Include Palm OS Global Find Support	37
Databases To Search – All Databases	38
Databases To Search – Select Databases...	38
Available DB's	38
Selected DB's	38
Remove Database	38
10 – Generating Code	39
Generated Code Sections	40
Database Structures	40
General Variables	40
Database Library (Header File)	40
Database Library (Code File)	40
App Start/Stop	41
Save Code As...	41
Save All	41
Copy To Clipboard	41
The Code Window	41
11 – Using da Vinci with CodeWarrior	42
Adding Files To Your CodeWarrior Project	42
Integrating Generated Files	43
Including Special Options	43
Example Code	43
12 – da Vinci Generated Functions Reference	44
Database Functions:	44
GetRecordPointers	44
PackRecord	44
UnpackRecord	45
GetRecord	45
PackedRecordSize	45
NewRecord	46
CreateNewDefaultRecord	46
SaveRecord	46
DuplicateRecord	47
IntDB	47
SortDatabase	48
Preference Database Functions:	48
GetPreference	48
SavePreference	48
Global Find Functions:	49
SaveGFPreference	49
GetGFPreference	49

StrFindIn	50
GlobalFind	50
13 – da Vinci Support	51
Web and Email Support	51
Phone Support	51
CodeWarrior Support	51
ETS Inc. Contact Information	51

User's Guide Overview

The da Vinci User's Guide is designed to help you get the most out of using da Vinci – The Data Architect. The following list describes the contents of each chapter.

Chapter	Contents
1 - Welcome to da Vinci	What is da Vinci and how you can benefit by its use?
2 - Getting Started	PC Requirements and Installation Instructions.
3 - Palm OS Databases	A brief overview of Palm OS databases and why da Vinci is the best approach.
4 - da Vinci Tutorial	A step by step lesson on using da Vinci.
5 – The da Vinci User Interface	A description of the toolbars and buttons.
6 - Creating and Managing a Project	How to create and manage a da Vinci Project.
7 - Creating and Managing Databases	How to create and manage a Palm OS Databases.
8 - Creating and Managing Fields	How to create and manage fields within the Palm OS Database.
9 - Project Properties	How to set the project properties.
10 - Generating Code	Generating and viewing code.
11 - Using da Vinci with CodeWarrior	How to utilize da Vinci generated code with CodeWarrior.
12 – da Vinci Generated Function Reference	Reference guide for all da Vinci generated functions.
13 - da Vinci Support	How to get technical support for using da Vinci.

1 – Welcome to da Vinci, the Data Architect

What is da Vinci?

da Vinci is a Database design and code generation tool for use with Metrowerks CodeWarrior for Palm OS (version 8 or later). The target market for da Vinci is Palm OS software developers, who do their software development with the C programming language.

The Birth of da Vinci

One of the most complex aspects of Palm OS development is the design and coding of database support for a Palm OS application. This became very evident to us when we started designing Palm OS applications in 1999. We soon discovered that a simple database design could take days to implement. Compared to the tools available for PC software development, we found it hard to believe that databases for Palm OS were so difficult.

In order to have the best database performance in the least amount of memory space possible, it was necessary to custom code for each individual database table and structure. This also meant custom database management functions for handling each database. The first logical step was to copy code from prior applications. But then this led to a lot of code changes after the "paste", and of course, the introduction of "copy/paste" errors.

After much discussion with other Palm OS developers, we determined that we were not the only developers suffering with the complexities of database code. After much searching, we discovered a complete lack of tools for doing database design. Thus the concept behind da Vinci was born.

We next came up with the criteria for a database design tool.

- It had to be easy to use.
- It had to generate efficient C code.
- It would not only generate database structure code but also generate database management functions.
- It would allow us to go back and make changes.

Now that's not asking for much, is it? Well, 2 years later, da Vinci is here!

Why Use da Vinci?

Here's the equation:

Time = \$

Database development takes time, and thus takes \$.

A simple database with 5+ fields can take hours to code.

A complex database with multiple databases, with lots of fields can take days and sometimes weeks to code

A developer using da Vinci can develop a simple database and all the necessary code in under 5 minutes. A highly complex database can be done in less than an hour.

The result is a savings of hundreds to thousands of dollars in development cost, per project, and a highly efficient application, using the raw power and performance of C code.

Why Use da Vinci Instead of A Database Engine?

The answer to this one is easy – Raw Power and Raw Speed!

Sure, there are database engines for Palm OS. However, they require the use of “plug-in” library technology, this means you’re using more memory than required which results in poorer performance. When the performance is bad there is nothing that can be done to “tweak” it in code, because you’re dependent upon the library routines that someone else wrote.

Hands down, the fastest and most efficient databases for Palm OS are all designed and developed with C code.

How Efficient Is the Code from da Vinci?

During the years that we have been doing Palm OS development, we have determined that most of the Palm OS programming examples from books, sample code, and other sources, is inefficient. The code is there to teach the concepts of Palm OS programming, but in most cases, does not address the need for performance or compactness.

Our developers have put forth a lot of effort and many a late night, trying to squeeze the most performance and efficiency out of Palm OS databases. While the generated code may not look like any examples you’ve ever seen, trust us – it works. and if you want to learn what the code is doing, just turn on the automatic commenting and each line of code will have a short comment about what is going on.

Will da Vinci Write a Complete Application?

No. While this might become a feature in future versions of da Vinci, currently you still have to write your own code for resources, forms, events, and other necessary functions.

2 – Getting Started with da Vinci, The Data Architect

Minimum PC Requirements

- Intel Pentium[®] or comparable, 100+ MHz, 64 MB RAM
- Microsoft Windows[®] NT 4.0 (Service pack 4 or later), 2000, XP Home, or XP Professional
- 20 MB of available hard disk space
- Metrowerks Inc. CodeWarrior for Palm OS, version 8.0 or later.
- Palm OS 3.1 or later.
- Palm OS 5.0 is not yet supported.

Recommended User Prerequisites

Though da Vinci makes it easy to design databases for Palm OS, it does require a significant amount of knowledge for using the generated code with CodeWarrior. It is recommended that the da Vinci user has the following skills:

- An understanding of how to use Metrowerks CodeWarrior for Palm OS, version 8.0 or later.
- A basic understanding of the C programming language.
- An understanding of how to develop Palm OS applications using the C programming language.

If you're just getting started with Palm OS programming, then Metrowerks has an excellent, on-line training program for learning Palm OS development with CodeWarrior. You can find out more by going to the following Metrowerks web site:

<http://www.codewarrioru.com/>

Installing da Vinci

Electronic Distribution Version

- If you have already downloaded the electronic distribution file, then you can just execute the EXE file from a temp directory on your local hard drive.
- Follow the Installer instructions.

CD ROM Distribution Version

- Insert the da Vinci CD-ROM into your PC.
- If your PC automatically runs CDs, then the da Vinci Installer will open. If it does not open, click Start, then Run, then type D:\SetupLeo.exe, where D is your CD-ROM's drive letter designation.
- Just follow the Installer instructions.

3 – Palm OS Databases

One of the reasons that coding databases for Palm OS is so complex is due to the fact there is no file system. Palm OS utilizes memory for database storage. There are no hard drives, and thus, no need for a file system. While this may change in the near future, currently we are limited to the use of memory for storing data.

Memory on Palm OS is divided into two areas: Dynamic memory and Permanent storage memory.

Dynamic memory is where your application and the OS operate while running. This is also where heap memory resides. Just like the RAM on your PC, when the Palm is reset, this memory is cleared.

Permanent memory is where applications and data are stored. Similar to the hard drive on your PC, this memory is not cleared during a reset.

The complexity of writing code for storing data into and retrieving data from permanent memory is due to the protection measures that Palm has implemented into the OS. This is done to prevent programmers from accidentally rewriting memory and thus losing vital data. Just imagine how users would feel if one badly behaved application wrote over the data for all their other applications. So, this memory can be considered as write protected. In order to write to the permanent memory area, special API calls have to be utilized. And even these have restrictions on the chunk size for writing and reading data.

This is why the concept of "Packing" and "Unpacking" is a standard for storing and reading data. True, you could just write raw data to permanent memory, but this would be inefficient and sloppy; (And would not conform to the Zen of Palm Programming.)

To utilize a database record, you have to copy it to dynamic memory, make your changes, and then write it back to permanent memory. Sounds simple, yes? Well, you also have to consider that Palm OS maintains key information about each record, the location of the record, a 3 byte unique identifier, and a 1 byte attribute indicator. Then there's the Sort Info block, and since we doing all this with the C programming language, we also have to maintain handles and pointers. Now you can see how this can quickly get complex, especially when you have to handle multiple databases.

An important terminology issue to consider with Palm OS is that each database is a single table of records. A database cannot have multiple tables in it.

If you want to learn more about Palm OS databases and writing code for databases then we recommend the following books:

- Palm Programming, The Developers Guide. By Neil Rhodes & Julie McKeenan. Published by O'Reilly. ISBN: 1-56592-525-4
- Palm Database Programming, The Complete Developer's Guide. By Eric Giguère. Published by Wiley Computer Publishing. ISBN: 0-471-35401-5

4 – da Vinci Tutorial

Quick Start

As a Palm OS developer, you know how difficult it can be to write all the necessary code for supporting a database application. With da Vinci, all the hard work is done for you. You design the database, the fields, and the properties for the database, and then da Vinci generates all the necessary code for using and managing the data.

In this tutorial, you'll step through the creation of a CD Music disk collection database. If you haven't installed da Vinci yet, then go through the Getting Started section first, then come back here to do the tutorial.

Now that you've installed da Vinci, you'll need a folder on your computer called "CDMusicApp". You'll be using this folder for storing the project and the generated code.

- ☐ Create a folder on your computer called "CDMusicApp". If your primary hard drive is C: then the path to your project folder should be C:\CDMusicApp
- ☐ Launch da Vinci

You'll now see the Main menu and toolbar for da Vinci.



Start a new database project called CD Database.

- ☐ Click on the New Project Wizard button. 

This will bring up the New Project Wizard window.

da Vinci Database Architect for PalmOS

New Project Wizard - Main Project Properties

Welcome to the da Vinci Database Architect New Project Wizard. This will lead you through a series of panels in helping you to create your project.

Please enter your basic project settings now. You can always change these later if you need to from the project properties form.

Project Name:
CD Database

Project Version: (For Preference Databases)
1

Project CreatorID: (4 Letters, Must Be Registered With Palm)
CDId

< Prev Next > Cancel

Enter main project properties. New Project Wizard

- ☐ Enter "CD Database" for the Project Name.
- ☐ Enter "1" for the Project Version.
- ☐ Enter "CDId" for the Project Creator ID.

The Creator ID is a unique id that has been registered with Palm Inc. for use with a specific application. Each Palm OS application requires a unique Creator ID. If you plan on distributing your application to other Palm OS devices, then you should register for a unique Creator ID. This can be done via Palm Inc.'s web site at <http://www.palmos.com>.


The next step is to add databases to the project.

- ☐ Click on the "Next" button to go to the next Wizard form.

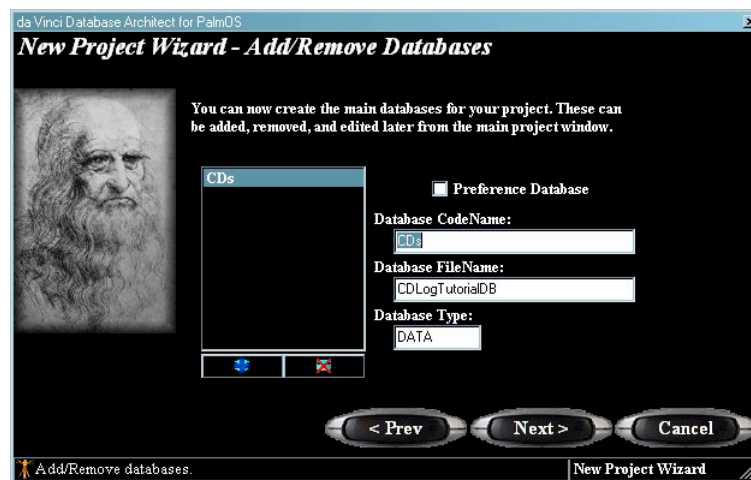


In this form you'll add a database to your da Vinci project.

- ❑ Click on the "Add Database" button. 

The "Remove Database" button  can be used to remove databases from the da Vinci project. Unless you make a mistake, you won't need it for this tutorial.

You should now see the Database detail fields.



- ☐ Enter "CDs" as the Database Code Name. The Database Code Name field is the name you'll use to refer to the database in the code. Generally it is good to use a short name.
- ☐ Enter "CDLogTutorialIDB" as the Database File Name. This is the name used by Palm OS for storing the database in memory.
- ☐ Enter "DATA" as the Database Type. This is a 4-letter identifier, and becomes very important if you decide to use the Open Database with Creator ID and Type as a code generation option. (Which you'll see later.)

This is all the information we need for the database itself. The next step is to add fields to the database.

- ☐ Click on the "Next" button to go to the next form.



- ☐ Select the "CDs" database from the drop down list.

The database field list box will now appear.



You can now use the "Add Field" button  and the "Remove Field" button  for adding and removing fields from the selected database.

- ❑ Click on the "Add Field" button.

You'll now see the attribute fields for the field you are adding to the database.






- ❑ In the Field Name box, type "CdsArtist."

- ☐ In the Field Type box, select "Char/Char*" from the list.
- ☐ Leave the Default Field Value box blank.

The Field Name is the value that you'll use for referencing the field in your code. The Field Type defines the type of data that will be stored in the field. Default Field Value allows you to define a default value to be added to the field every time a new record is added to the database.

Now add the other fields to the database:

- ☐ Click on the "Add Field" button .
- ☐ In the Field Name box, type "CdsTitle."
- ☐ In the Field Type box, select "Char/Char*" from the list.
- ☐ Leave the Default Field Value box blank.
- ☐ Click on the "Add Field" button .
- ☐ In the Field Name box, type "CdsYear."
- ☐ In the Field Type box, select "Char/Char*" from the list.
- ☐ Leave the Default Field Value box blank.
- ☐ Click on the "Add Field" button .
- ☐ In the Field Name box, type "CdsGenre."
- ☐ In the Field Type box, select "UInt16" from the list.
- ☐ Leave the Default Field Value box blank.

You have finished adding the fields to the database. The screen should now look like this:



Now you're ready to move on, saving the project and generating the code.

- ☐ Click on the "Next" button.

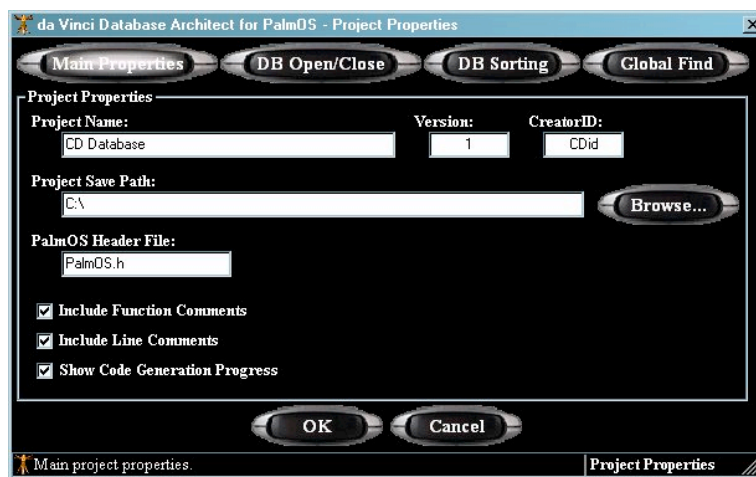
Congrats! You have finished the steps for creating the project.



For this project, you'll need to view the Advanced Settings and then Generate the source code.

- ☐ Make sure that all three (3) check boxes are checked.
- ☐ Then click on the "Finish" button.

You should now see the Project Preferences screen.

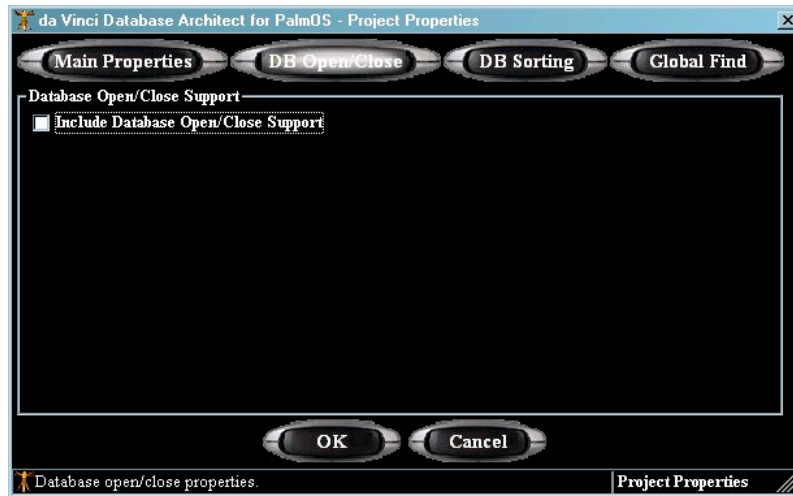


The settings for Project Name, Version, and Creator ID were set earlier when you started the Project Wizard. You can change the Project Save Path to the folder you created at the start of this tutorial: "C:\CDMusicApp"

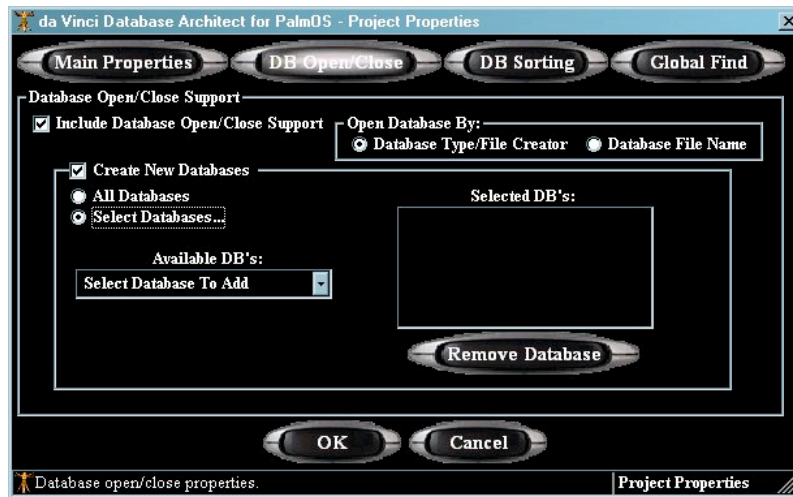
- ☐ Change the Project Save Path to "C:\CDMusicApp". (You can type this in directly or use the "Browse..." button.)

This project needs to have Database Open and Close Support.

- ☐ Click on the "DB Open/Close" button.



- ☐ Click the "Include Database Open/Close Support" check box.

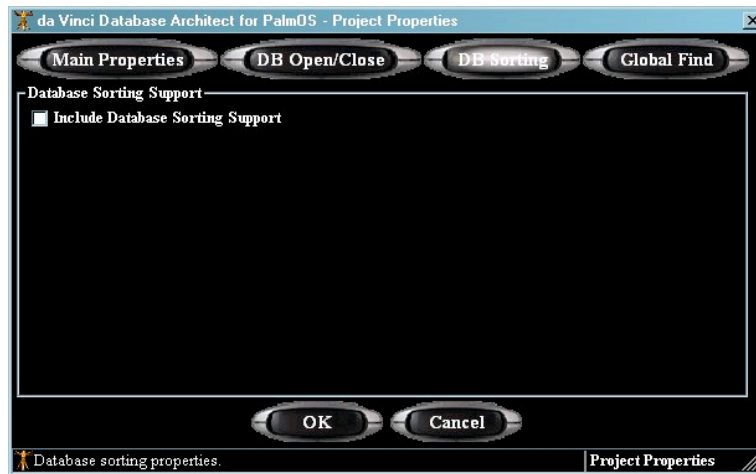


- ☐ Make sure that the "Database Type/File Creator" button is selected.
- ☐ Make sure the "Create New Databases" checkbox is checked.
- ☐ Click on the "Select Databases..." button.
- ☐ Select the "CDLogTutorialDB" from the list.

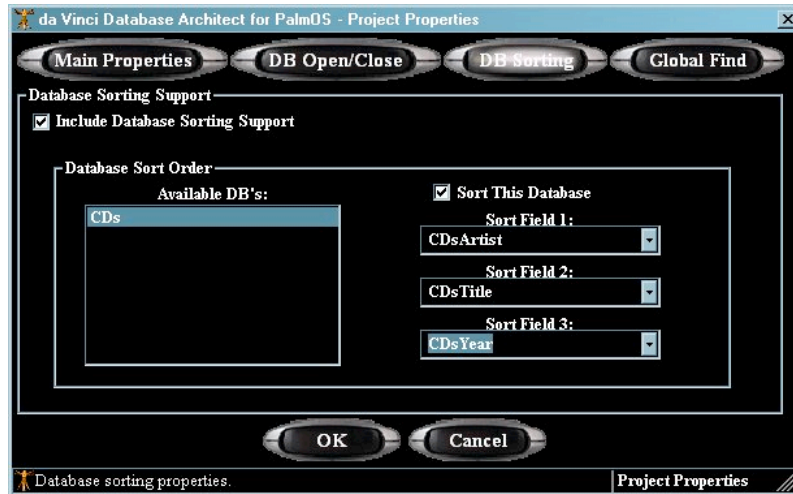
You should see the selected database appear in the "Selected DB's:" window.

da Vinci can also generate the necessary code for sorting.

- ☐ Click on the "DB Sorting" button.



- ❑ Click the "Include Database Sorting Support" check box.



You'll now see the sorting options available for your database. The sample application will use three (3) of the fields for sorting.

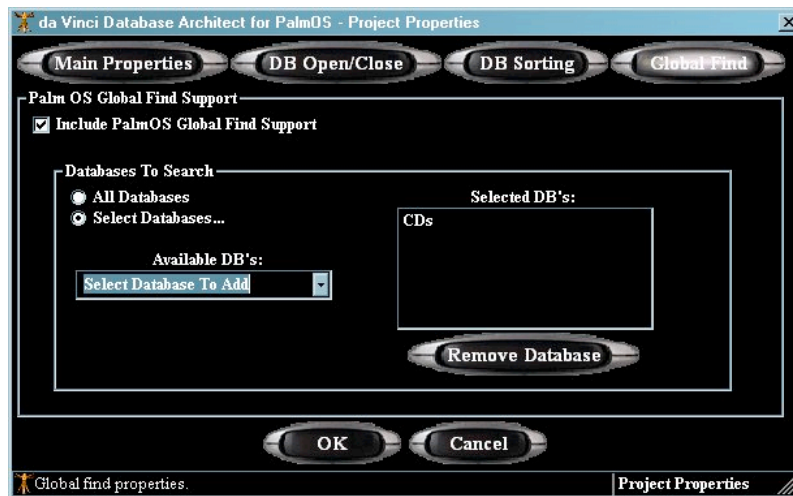
- ❑ Make sure that the "CDs" database is highlighted in the "Available DB's" list box.
- ❑ Click the "Sort This Database" check box.
- ❑ Set Sort Field 1 to the "CDsArtist" field.
- ❑ Set Sort Field 2 to the "CDsTitle" field.
- ❑ Set Sort Field 3 to the "CDsYear" field.

You'll also want to utilize the Global Find capabilities of Palm OS for the application.

- ❑ Click the "Global Find" button.



- ☐ Click the "Include Palm OS Global Find Support" check box.

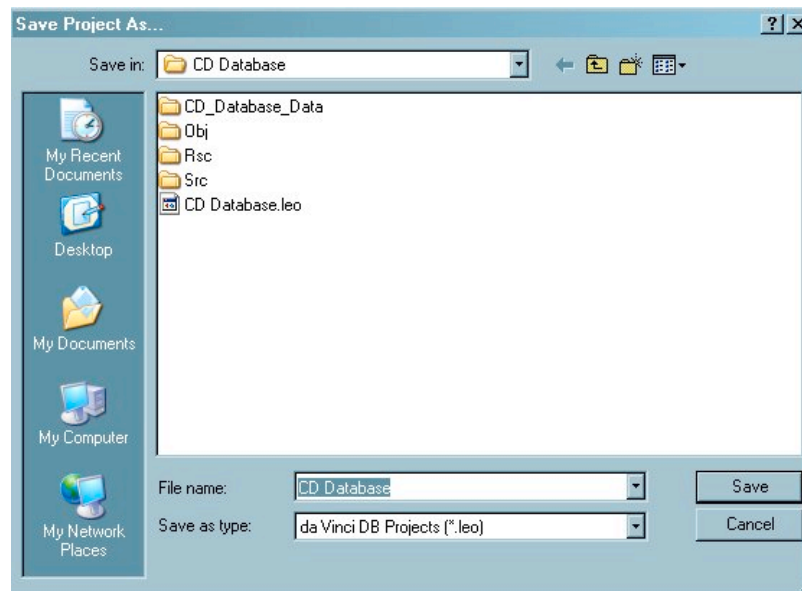


- ☐ Click the "Select Databases..." button.
- ☐ Select the "CDs" database from the "Available DB's" list box.

Now you're all set to generate the code for your project.

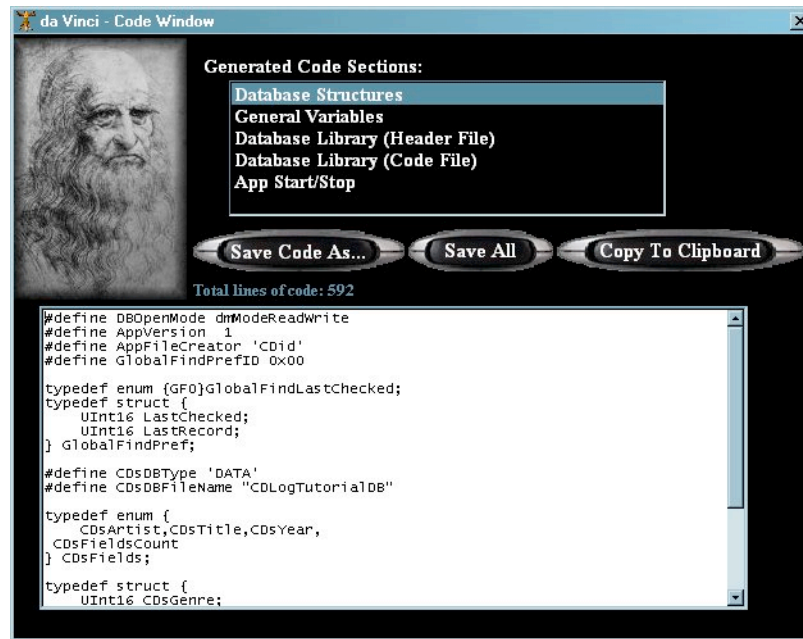
- ☐ Click on the "OK" button.

You'll now be prompted to save the project.



- ☐ Make sure that the File name is "CD Database" then click on the "Save" button.

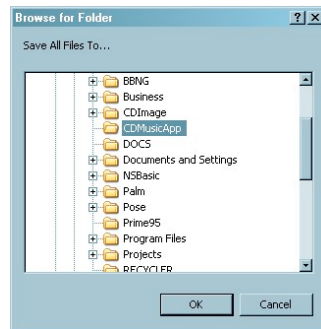
Depending on the speed of your computer, you may or may not be able to see the progress bar form while the source code is being generated. Once the code is generated, you'll be able to view each of the code segments.



As you're reviewing the generated code, you will also have the ability to make changes. But, it is best to wait and make changes to the generated code after it has been added to your CodeWarrior project.

You can add da Vinci's generated code by either: copying to the clipboard, and then pasting into CodeWarrior, (you can use the "Save Code As..." button to save each individual file, or you can use the "Save All" button.) For the purpose of this tutorial you'll use the "Save All".

- ☐ Click on the "Save All" button.



- ❑ Be sure to select the correct folder for your project, and then click on the "OK" button.

Next, you'll see a confirmation that all the files have been saved. You can now close the Code Window.

- ❑ Click on the "X" icon in the top right corner of the Code Window.

If you look at the file list for your project folder, you should see the following files:

- AppStartStop.txt
- CD Database.leo
- DBLibrary.c
- DBLibrary.h
- DBStructures.h
- Variables.h

The AppStartStop.txt file contains instructions and some code that needs to be pasted into your CodeWarrior project.

The .c and .h files can be directly added to your CodeWarrior project. If you want to see an example of how these modules are utilized in a fully functional CodeWarrior application, then just open the \Examples\Tutorial\CD Database\CD Database.mcp file into your local copy of CodeWarrior. This is a fully functional application written with da Vinci and CodeWarrior.

Congrats! You are now on your way to being an expert database programmer for Palm OS.

5 – The da Vinci User Interface

The primary interface for da Vinci is the Main Toolbar and Main Menu.

da Vinci Toolbar



When there is no active project the following buttons are active:



New Project Wizard – Starts the New Project Wizard for stepping through the creation of a new project, adding databases, fields, and generating code.



New Project – Creates a new blank project and displays the Add Database form.



Open Project – Displays the Open File Dialog box for selecting an existing da Vinci project.

This is the toolbar for an Active project:



When there is an active project, the following buttons are active:



Save Project – Saves the current da Vinci project. If the project has not yet been saved, the "Save As Dialog" box will display. This will allow you to select the path and filename for the new project.



Add Database – Adds a new database to the current project.



Add Field – Adds a new field to the current database.



Remove Database – Removes the currently selected database from the current project.



Remove Field – Removes the currently selected field from the current database.



Project Properties – Displays the Project Properties form for the current project.



Generate Code – Generates source code from the currently selected project.

da Vinci Menus


There are three (3) primary menus for da Vinci – File, Options, and Help. Here are the available menu selections under each:

Menu	Sub Menu	Description
File	New Project	Start a new project
	Open Project	Open an existing project
	Close Project	Close the current project
	New Project Wizard	Use the New Project Wizard
	Save Project	Save the current project
	Save Project As	Save and rename the current project
	Exit	Exit from da Vinci
Options	Add Database	Add a database to the current project
	Add Field	Add a field to the current database
	Remove Database	Remove the selected database from the current project
	Remove Field	Remove the selected field from the current database
	Project Properties	Display the current project properties
Help	Generate Database Code	Generate source code for the current project
	About da Vinci	Display the version, credits, and rights for da Vinci

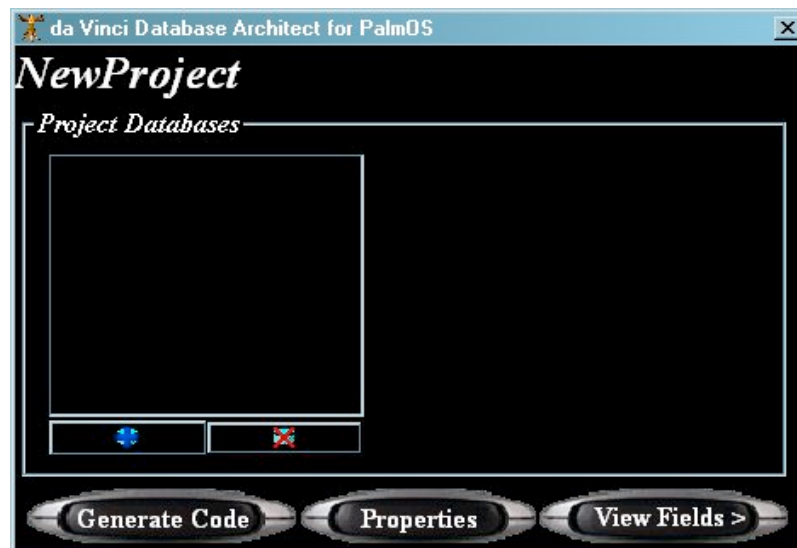
6 – Creating and Managing a Project

Creating and managing projects with da Vinci is a very easy process. When a project is created with da Vinci, a project file with a file extension of “.leo” is created in the selected project directory. **Please note: Although this file is in a readable text format, it is very important that you not make any changes to this file outside of da Vinci. To do so, could make this file unreadable by da Vinci.** da Vinci uses this project file to store vital information about the project, the databases, and the properties of the project.

Creating a Project

To create a new project in da Vinci, just select the New Project icon  from the tool bar, or you can select the “File” menu and then select the “New Project” selection. (A key sequence of Ctrl & N can also be used.)

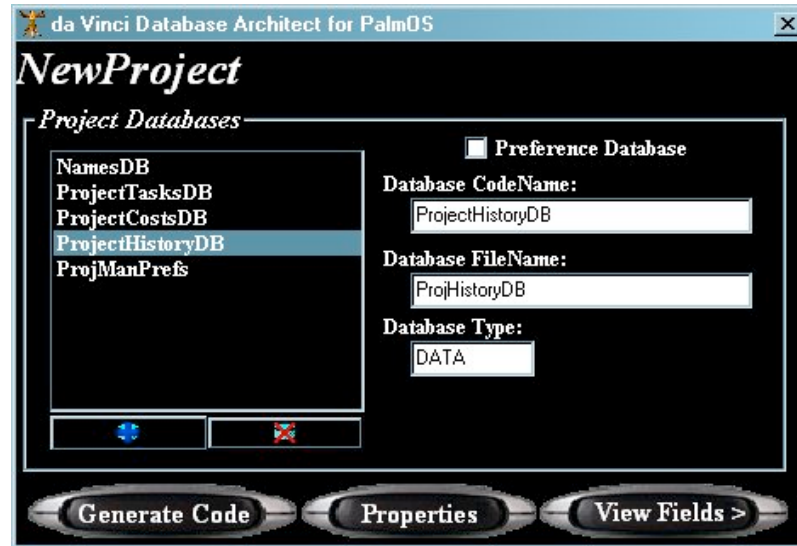
The New Project window will appear with an empty Project Database list box.



Only one project can be open in da Vinci at a time. Each project can have a maximum of 20 databases, and each database can have up to 200 fields.


Currently, da Vinci does not support the import or export of database structures, as this feature is currently under design for inclusion in future versions of da Vinci.

7 – Creating and Managing Databases




Functions

New Database

Select the "Add Database" button  at the bottom of the Project Databases list box or from the da Vinci main toolbar. You can also select the "Options" menu, then the "Add Database" selection.

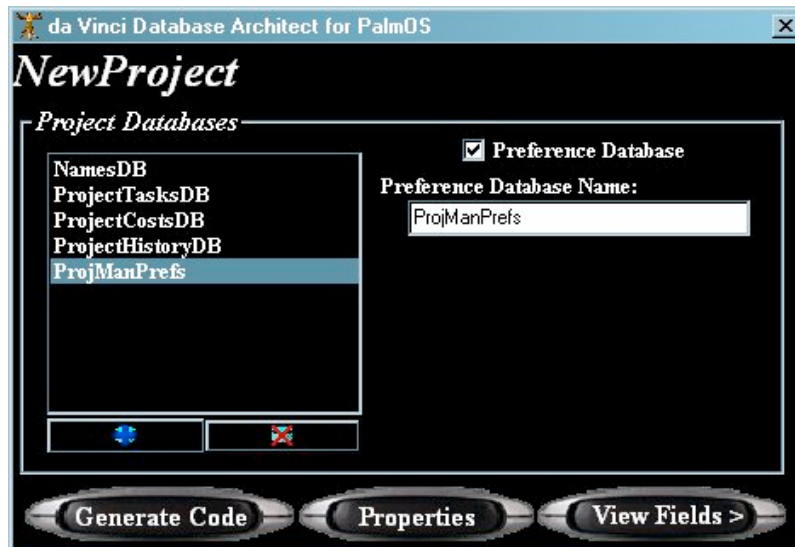
Remove Database

Select the "Remove Database" button  at the bottom of the Project Databases list box or from the da Vinci main toolbar, or select the "Options" menu, then the "Remove Database" selection.

There is no "save" button or feature. Changes are updated when the New or Remove functions are used. If you add an extra database by accident, then you can delete it with the "Remove" function.

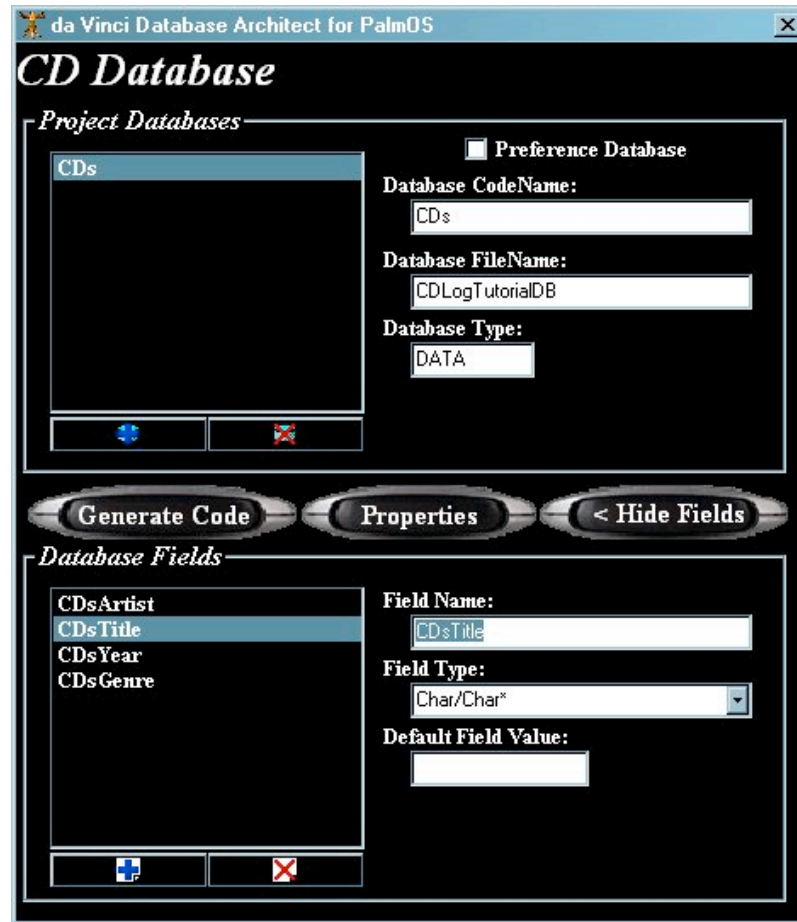
Preference Database

The "Preference Database" checkbox identifies that the currently selected database is a Palm OS application Preference Database. Database Filename and Database Type are not needed for these types of databases.




A scroll bar will appear on the right side of the Project Databases list box if there are more databases than the visible list can show.

8 – Creating and Managing Fields




Functions

Add a Field

Select the "Add Field" button  at the bottom of the Database Fields list box or from the da Vinci main toolbar, or select the "Options" menu, then the "Add Field" selection.

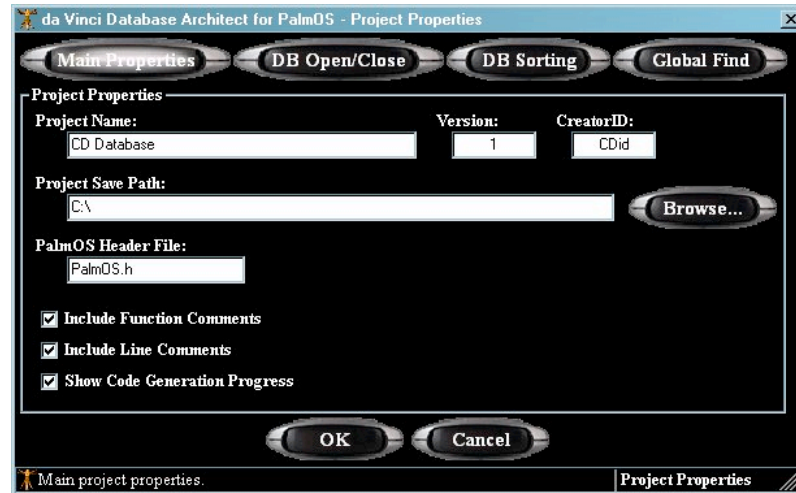
Remove a Field

Select the "Remove Field" button  at the bottom of the Database Fields list box or from the da Vinci main toolbar, or select the "Options" menu, then the "Remove Field" selection.

There is no "save" button or feature. Changes are updated when the New or Remove functions are used. If you add an extra field by accident, then you can delete it with the Remove function.

A scroll bar will appear on the right side of the Database Fields list box if there are more fields than the visible list can show.

9 – Project Properties



Main Properties Fields and Functions

Project Name

This is the name of the project used for storage on your PC. Although it does not show on this form, the file name will have an extension type of ".leo"

Version

This is for recording a version number associated with your project, and is purely for your personal version tracking needs and does not have any affect upon the generation of code.

CreatorID

The Creator ID is a unique id that has been registered with Palm Inc. for use with a specific application. Each Palm OS application requires a unique Creator ID. If you plan on distributing your application to other Palm OS devices then you should register for a unique Creator ID. This can be done via Palm Inc.'s web site at <http://www.palmos.com>.

Project Save Path

This is the path on your local PC where the project file is saved. The contents can be typed in directly or you can select the appropriate path with the "Browse..." button.

Palm OS Header File

This is the name of the Palm OS header file that you are using for your CodeWarrior project, and has to be included in the generated source code.

Include Function Comments

If checked, comments will be added to the generated source code for the Functions.

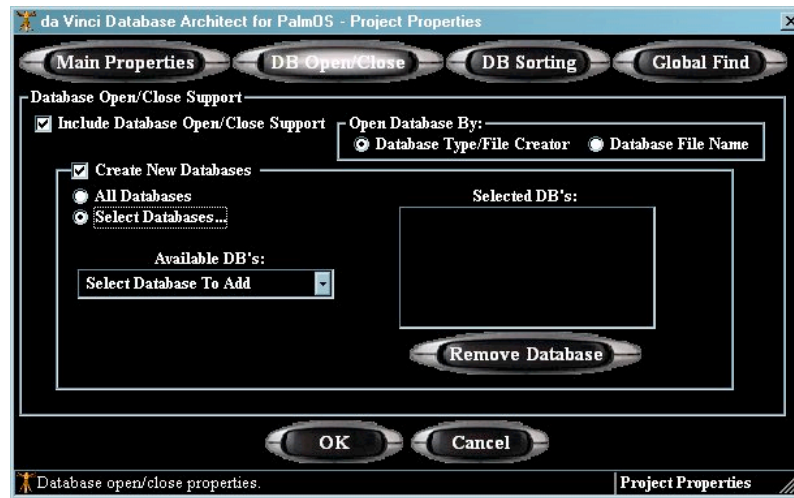
Include Line Comments

If checked, comments will be added to the rest of the generated source code.

Show Code Generation Progress

If checked, a progress bar will show code generation progress. (Not really necessary unless you're using a slow computer and generating code for a large project.)

DB Open/Close Properties



Database Open/Close Support

When checked, da Vinci will generate the appropriate source code for database "Open and Close" functions. Various code generation options (see the following topics) become available if this checkbox is checked.

Open Database By

These two options – "Database Type/File Creator" or "Database File Name", determine how the database Open/Close function code is generated. If you want to Open and Close your databases by Database Type and File Creator, then you will want to select the button for that option.

Create New Databases

When this option is checked, da Vinci will generate the necessary code functions for creating the databases. This provides the capability to automatically create the databases the first time your application is run on the Palm device.

All Databases

If this option is selected, da Vinci will generate the "create database" function code for all databases in the project.

Select Databases...

With this option, da Vinci will generate the "create database" function code for only the selected databases.

Available DB's

This list box shows all the available databases in the current project. Databases can be selected from this list for inclusion as databases to have the "create database" function code generated. The selected databases will appear in the "Selected DB's" box.

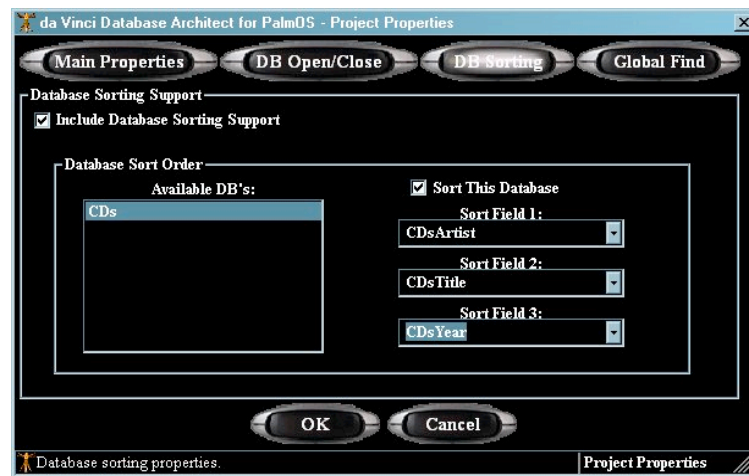
Selected DB's

This box shows a list of databases that have been selected for generating the "create database" function code.

Remove Database

This will remove the currently selected database from the "Selected DB's" box.

DB Sorting Preferences



Include Database Sorting Support

When checked, da Vinci will generate the appropriate source code for database sorting functions. Various code generation options, (see following topics) become available if this checkbox is checked.

Available DB's

This list box shows all the available databases in the current project. The currently selected database will have its fields available for selection as sort fields.

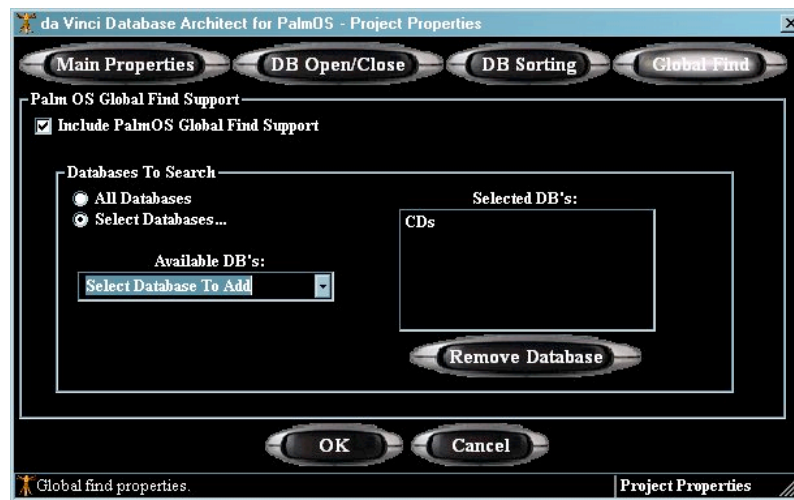
Sort This Database

When selected, the currently selected database will have its fields available as sort fields.

Sort Field(s)

The fields from the currently selected database are available in these dropdown list boxes. Up to three (3) fields can be selected as sort fields. da Vinci will generate sort functions for the selected fields.

Global Find Preferences



Include Palm OS Global Find Support

When checked, da Vinci will generate the necessary function code for providing "Global Search" capabilities for the selected databases.

Databases To Search – All Databases

If checked, da Vinci will generate "Global Search" code for all of the databases in the project.

Databases To Search – Select Databases...

When selected, you'll be able to select which databases to include for the generated "Global Search" code.

Available DB's

This list box shows all the available databases in the current project. Databases can be selected from this list for inclusion as databases to have the "Global Search" function code generated. The selected databases will appear in the "Selected DB's" box.

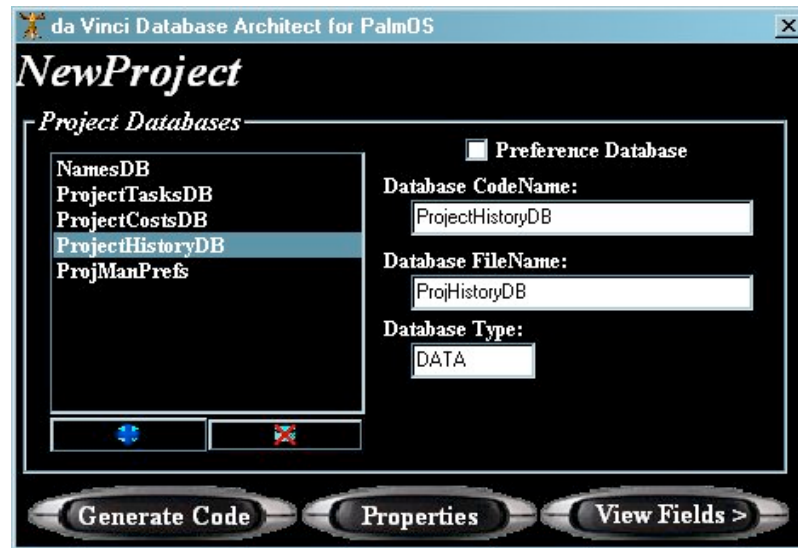
Selected DB's

This box shows a list of databases that have been selected for generating the "Global Search" function code.

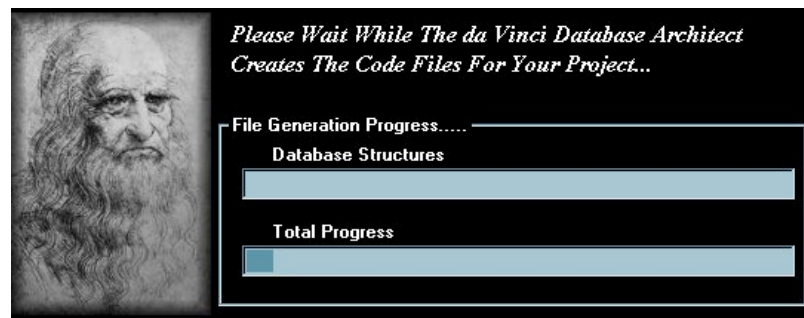
Remove Database

This will remove the currently selected database from the "Selected DB's" box.

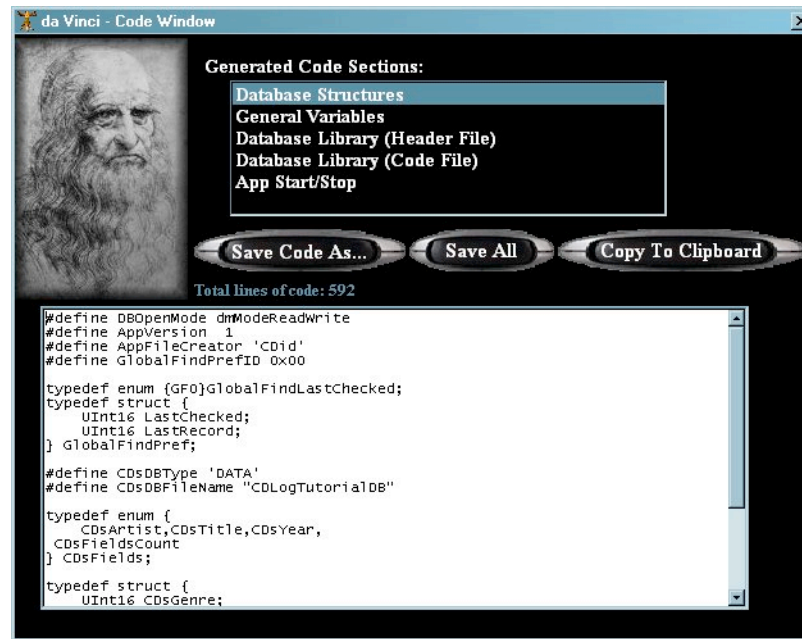
10 – Generating Code



Source code is generated by clicking on the "Generate Code" button.



After the code is generated, the Code View window is displayed.



Generated Code Sections

This is a list of the code modules that were generated by da Vinci. The highlighted code module is the source of the text in the Code Window.

Database Structures

This code segment contains all of the database structure definitions. When saved, the default filename is "DBStructures.h."

General Variables

This code segment contains all of the Variable definitions for the generated code. The default filename for this module is "Variables.h"

Database Library (Header File)

The code in this module is a header file with all the references for the generated functions. The default filename for this module is "DBLibrary.h." (See the section called "da Vinci Generated Functions Reference" for details on how to use these routines.)

Database Library (Code File)

This contains the functions and routines for managing the database within your application. The default filename for this module is "DBLibrary.c." (See the section

called "da Vinci Generated Functions Reference" for details on how to use these routines.)

App Start/Stop

This contains code segments that need to be placed into various locations in your CodeWarrior application. Each code segment begins with comments that give clear instructions on where the code needs to be placed. (See the chapter on "Using da Vinci with CodeWarrior" for more details.)

Save Code As...

This will allow you to save each code section with a custom filename.

Save All

This will save all the code sections with a default naming convention. The default filenames are:

- AppStartStop.txt
- DBLibrary.c
- DBLibrary.h
- DBStructures.h
- Variables.h

Copy To Clipboard

This will place the contents of the currently selected code section into the Clipboard. Once in the Clipboard, the code can then be "pasted" into CodeWarrior. **(Please note: The code text will not appear correctly if you try to paste it into Notepad or WordPad. da Vinci has been designed for use with CodeWarrior. Use with other editors is not supported.)**

The Code Window

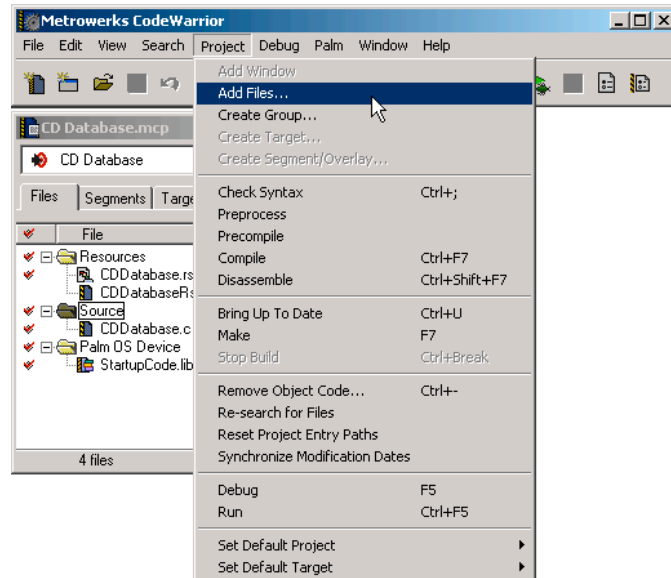
The scrollable Code Window is primarily designed for viewing of the generated code. The code can be edited in this window, and the code changes will be saved or copied.

11 – Using da Vinci with CodeWarrior

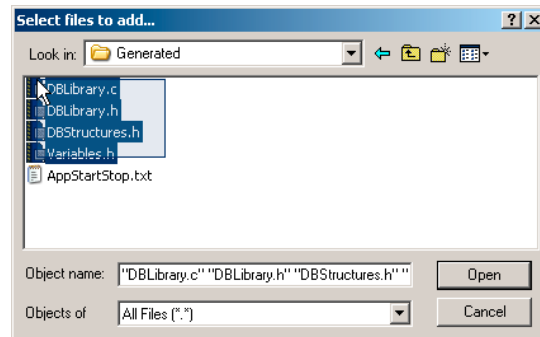
Adding Files To Your CodeWarrior Project

Once the code for your database(s) have been generated, all that remains is adding these files to your CodeWarrior project. This section will step through the process needed to successfully add generated files to any CodeWarrior project.

Make sure that your project is open in CodeWarrior. Then use the 'Project' menu to add files to your project as shown here:



Select your generated code files and click on the "Open" button.



Integrating Generated Files

Now that all of your generated code files have been added to the CodeWarrior project, you need to integrate them with each other and all other files that will be using them. In order to do this you first need to include the variables at the top of our main .c file (ex: #include "Variables.h"). Next, for each .c file that will call any of the functions that exist in the generated database library (including the database library file itself) or use the database for which code was generated, you need to include 2 files.

The first file is the generated structures file (ex: #include "DBStructures.h"), which should be included at the top of the .c file.

Next you need to include the function definitions and database globals for the database library (ex: #include "DBLibrary.h"). This should be included right after the structures file.

After these files are included you will be able to call the functions in the generated database library. For help on using these functions please refer to the da Vinci Generated Function Reference section.

Including Special Options

If you included some of the special options such as the 'Database Open/Close Support' or the 'PalmOS Global Find Support' you may have an extra file generated, which by default is named "AppStartStop.txt". In this file are some further instructions on how to incorporate these special options into your CodeWarrior project.

Example Code

If you need additional help on how to reference the databases with your code, just take a look at the example application that comes with da Vinci. Look in the following folder

\Program Files\daVinci\Examples\Tutorial\CD Database

for the "CD Database.mcp" CodeWarrior project file. Open this up in CodeWarrior. This project is heavily commented and is a good example of how to use da Vinci generated code.

12 – da Vinci Generated Functions Reference

Database Functions:

The main database functions are included for the purpose of doing basic things such as saving records, creating new records, and opening records. Some of the following functions require certain preferences to be enabled.

GetRecordPointers

Purpose: Returns the pointers to the start of the string fields for a database structure.

Prototype: void GetRecordPointers(DmOpenRef dbP, void *item, void *recordP, Char **s1, Char ***s2, Int32 *x)

Parameters:	-> dbP	Database open reference of the database structure to get.
	-> item	Pointer to the unpacked database structure.
	-> recordP	Pointer to the packed database structure.
	<- s1	Pointer to the string to have the start of the strings field of the packed structure.
	<- s2	Pointer to the string to have the start of the strings field of the unpacked structure.
	<- x	Pointer to an integer to be filled with the number of fields in the structure.

Result: Returns nothing.

Comments: To avoid getting the pointers for either 'item' or 'recordP' pass in a NULL.

PackRecord

Purpose: Packs a database record.

Prototype: void PackRecord(DmOpenRef db, void *item, void *recordP)

Parameters:	-> db	Database open reference of the database the record belongs to.
	-> item	Pointer to the unpacked database structure.
	<- recordP	Pointer to the packed database structure.

Result: Returns nothing.

Comments: Packs the record by merging all of the strings into 1 string separated by NULL's. (For easier storing)

UnpackRecord

Purpose: Unpacks a database record.

Prototype: void UnpackRecord(DmOpenRef db, void *item, void *src)

Parameters:	-> db	Database open reference of the database the record belongs to.
	<- item	Pointer to the unpacked database structure.
	-> src	Pointer to the packed database structure.

Result: Returns nothing.

Comments: Unpacks the record by separating the single string into their respective strings.

GetRecord

Purpose: Gets a record from a database.

Prototype: Err GetRecord(DmOpenRef db, UInt16 recordNum, void *item)

Parameters:	-> db	Database open reference of the database the record belongs to.
	-> recordNum	Record number of the record to get.
	<- item	Pointer to the unpacked database structure to receive the data.

Result: dmErrIndexOutOfRange, or 0 if there was no error.

PackedRecordSize

Purpose: Returns how big a packed record would be.

Prototype: UInt32 PackedRecordSize(DmOpenRef db, void *item)

Parameters:	-> db	Database open reference of the database the record belongs to.
	-> item	Pointer to the unpacked database structure holding the data.
Result:	How big a packed record containing this data would be.	

NewRecord

Purpose: Creates a new record for the specified database.

Prototype: void NewRecord(DmOpenRef db, void *item, UInt16 *index)

Parameters:	-> db	Database open reference of the database the record belongs to.
	-> item	Pointer to the unpacked database structure holding the data.
	<- index	Record number where the new record is.
Result:	Returns nothing.	

CreateNewDefaultRecord

Purpose: Creates a new record for the specified database with all of the default values.

Prototype: void CreateNewDefaultRecord(DmOpenRef db)

Parameters:	-> db	Database open reference of the database the record belongs to.
Result:	Returns nothing.	

Comments: Sets the current record for the specified database to the record number of the new record.

SaveRecord

Purpose: Saves a record to the specified database.

Prototype: void SaveRecord(DmOpenRef db, void *item, UInt16 RecNum)

Parameters:	-> db	Database open reference of the database the record belongs to.
	-> item	Pointer to the unpacked database structure holding the data.
	-> RecNum	Record number of the record to replace with this one.
Result:	Returns nothing.	

DuplicateRecord

Purpose: Duplicates a record in the specified database.

Prototype: void DuplicateRecord(DmOpenRef db, UInt16 recNum)

Parameters:	-> db	Database open reference of the database the record belongs to.
	-> recNum	Record number of the record to duplicate.
Result:	Returns nothing.	

Comments: Sets the current record for the specified database to the record number of the new duplicate record.

IntDB

Purpose: Opens a database.

Prototype: void IntDB(DmOpenRef *dbPP, const Char* DBName, UInt32 DBType)

Parameters:	<- dbPP	Database open reference of the newly opened database.
	-> DBName	Name of the database to open.
	-> DBType	Type of the database to open.
Result:	Returns nothing.	

Comments: Only available when including the 'Database Open/Close Support'

SortDatabase

Purpose: Sorts a database.

Prototype: void SortDatabase(DmOpenRef dbP)

Parameters:	-> dbP	Database open reference of the database to be sorted.
-------------	--------	---

Result: Returns nothing.

Comments: Only available when including the 'Database Sorting Support'

Preference Database Functions:

Preference database functions are included for the purpose of saving and loading application preferences. All preference database functions are only available when there is at least one preference database included in the generated project.

GetPreference

Purpose: Gets the specified application preference.

Prototype: void GetPreference(UInt16 prefID, void *item)

Parameters:	-> prefID	ID of the preference to get.
	<- item	Pointer to the preference database structure to receive the data.

Result: Returns nothing.

Comments: If the specified preference was not found, it is created with the default values.

SavePreference

Purpose: Saves the specified application preference.

Prototype: void SavePreference(UInt16 prefID, void *item)

Parameters:	-> prefID	ID of the preference to save to.
	-> item	Pointer to the preference database structure holding the data.
Result:	Returns nothing.	

Global Find Functions:

Global find functions are included for the purpose of making it easier to have applications that use this feature provided by Palm OS. The included global find functions are designed to support searching through multiple databases and remembering which database was last searched. All global find functions are only available when including the 'Palm OS Global Find Support'.

Global Find Structures:

```
typedef struct {
    UInt16 LastChecked;
    UInt16 LastRecord;
} GlobalFindPref;
```

SaveGFPreference

Purpose: Saves the global find preferences.

Prototype: void SaveGFPreference(GlobalFindPref *GFPref)

Parameters:	-> GFPref	The global find preference structure containing the data to be saved.
-------------	-----------	---

Result: Returns nothing.

GetGFPreference

Purpose: Gets the global find preferences.

Prototype: void GetGFPreference(GlobalFindPref *GFPref)

Parameters:	<- GFPref	The global find preference structure to be populated.
-------------	-----------	---

Result: Returns nothing.

StrFindIn

Purpose: Finds one string inside another for global find purposes.

Prototype: Boolean StrFindIn(Char *strSearch, Char *strFind, UInt32 *pos, UInt16 *length)

Parameters:	-> strSearch	The string to search in.
	-> strFind	The string to search for.
	<- pos	Starting position of the found string.
	<- length	Length of the found string

Result: Whether the string was found or not.

Comments: "The search that TxtFindString performs is local-dependent. On most ROMs with Latin-based encodings, TxtFindString returns true only if the string is at the beginning of a word. On Shift-JIS encoded ROMs, TxtFindString returns true if the string is located anywhere in the word."
Palm OS Reference
This handles the cases where it doesn't search throughout the whole string.

GlobalFind

Purpose: Searches all of the databases for the specified string.

Prototype: void GlobalFind(FindParamsPtr findParams)

Parameters:	<- GFPref	The parameters for the search. Supplied by Palm OS through the cmdPBP parameter during a sysAppLaunchCmdFind launch code.
--------------------	-----------	---

Result: Returns nothing.

13 – da Vinci Support

Web and Email Support

ETS, Inc. will provide free support for da Vinci via our web based support forum at <http://www.exts.com>. Just go to our web site and click on the "Support" button. Once on our Support Board, go to the "da Vinci Support" forum. Our support staff monitors the board and will respond in as timely a manner as possible.

An ongoing FAQ for da Vinci is maintained on the web site and is available via the Support Board. We highly recommend that you download the latest version and check it whenever you have a question. Many of the questions and responses that we receive from other da Vinci users will be added to the FAQ. If you have a question and/or a usage tip, feel free to share it with us via the Support Board or via email.

Email support is available via the following email address: dVSupport@exts.com. While our support engineers will try to respond as quickly as possible, we feel confident that we can respond to your request within 8 business hours.

Phone Support

ETS, Inc. does not currently provide technical support via telephone. Please keep checking our web site for announcements on new technical support plans.

CodeWarrior Support

ETS, Inc. does not provide technical support for CodeWarrior usage. Please contact Metrowerks Inc. for your CodeWarrior support needs. We also recommend the use of the following newsgroups on the internet:

- codewarrior.palm
- pilot.programmer.codewarrior
- pilot.programmer

ETS Inc. Contact Information

ETS, Inc. (Extended Technology Systems, Inc.)
2531B State St.
Carlsbad, CA. 92008-1625
760-729-3396 Business office calls only
760-729-3397 fax
info@exts.com for general company information
dVSupport@exts.com for da Vinci technical support
<http://www.exts.com> Company web site

ETS, Inc. is a privately held corporation. Business, marketing, sales, and investment inquiries can be made via info@exts.com