



Developer's Guide For



950 Boardwalk, Suite 205, San Marcos, CA 92078 • (760) 510-1200 • www.productivecomputing.com
© Copyright 2010 Productive Computing, Inc.

Table of Contents

| | |
|---|----|
| 1) OVERVIEW | 3 |
| Purpose | 3 |
| Intended Audience | 3 |
| FileMaker Files | 3 |
| 2) RELATIONSHIPS..... | 4 |
| Figure 2.0 -Relationships for Core2 Contacts file | 5 |
| Figure 2.1 - Relationships for Core2 Documents file | 6 |
| Figure 2.2 - Relationships for Core2 Preferences file | 6 |
| Figure 2.3 - Relationships for Core2 Invoices file..... | 7 |
| Figure 2.4 - Relationships for Core2 Calendar file..... | 8 |
| 3) TABLE DESCRIPTIONS | 9 |
| Figure 3.1 - Core2 Contacts file | 10 |
| Figure 3.2 - Core2 Documents file | 12 |
| Figure 3.3 - Core2 Preferences file..... | 13 |
| Figure 3.4 - Core2 Invoices file..... | 14 |
| 4) FIELDS | 16 |
| 5) ENTITY RELATIONSHIP DIAGRAM | 17 |
| 6) SCRIPTS | 18 |
| 7) TOG LEGENDS..... | 19 |
| 8) USER ACCOUNTS AND FILE SECURITY..... | 21 |
| 9) CONTACT US | 22 |

1) Overview

Purpose

By popular demand we are proud to provide developers with an “under the hood” overview of our Core² CRM and Core² Invoices. The purpose is to not only provide you with a developer overview before you purchase the Core, but also to provide you with a navigational resource to assist you with your customizations after you purchase the Core. The possibilities to expand the Core are nearly endless. We would like to give you a better look at the files you will be customizing before and after your purchase.

Intended Audience

This Developer’s Guide is intended for FileMaker Developers only. Core users should refer to the Quick Start Guide and online videos at www.core2crm.com. It is recommended that FileMaker Developers be familiar with scripting, calculations, and relationships before making any “under the hood” changes to the Core files.

FileMaker Files

The purchased version comes to you fully unlocked and ready for customizations.
Files included in your package may vary.

The Core² CRM consists of the following files:

Core2 Contacts.fp7 = stores all contacts, activity and event records
Core2 Calendar.fp7 = maintains all calendar views, stores category and category colors
Core2 Documents.fp7 = stores all documents you attach to the contacts and to emails
Core2 Preferences.fp7 = stores all preferences, graphics, phone formats and user settings

The Core² Invoices consists of the following files:

Core2 Preferences.fp7 = stores all preferences, graphics, and user settings
Core2 Invoices.fp7 = stores all invoices, items, payments and QuickBooks support records

The Core² CRM with Invoices consists of the following files:

Core2 Contacts.fp7 = stores all contacts, activity and event records
Core2 Calendar.fp7 = maintains all calendar views, stores category and category colors
Core2 Documents.fp7 = stores all documents you attach to the contacts and to emails
Core2 Preferences.fp7 = stores all preferences, graphics, phone formats and user settings
Core2 Invoices.fp7 = stores all invoices, items, payments and QuickBooks support records

Figure 1.0 – Example of Core² CRM with Invoices files

| Name ▲ | Size | Type |
|---------------------|----------|--------------------|
| Core2 Calendar.fp7 | 1,272 KB | FileMaker Database |
| Core2 Contacts.fp7 | 1,720 KB | FileMaker Database |
| Core2 Documents.fp7 | 308 KB | FileMaker Database |
| Core2 Invoices.fp7 | 1,752 KB | FileMaker Database |
| Core2 Prefs.fp7 | 2,336 KB | FileMaker Database |

2) Relationships

All files in the Core² with the exception of the Calendar file are primarily based on the “anchor/buoy method.” In general, the theory is to base each layout of its own table occurrence and build relationships from left to right. Anchor/Buoy also incorporates relationship naming standards. It would take a much longer document to fully explain this method and that goes beyond the scope of this document. For more information, there is a great link you can read about this methodology:

<http://www.kevinfrank.com/anchor-buoy.html> (I recommend the “short” power point presentation).

Anchor/Buoy is used throughout the industry and has been the unofficial standard when working with FileMaker files in a multi-developer environment.

The Calendar file however is based on a traditional bi-directional relationship graph with naming standards that describe what the relationships is used for. This file was created before we adopted the anchor buoy method for the rest of the Core² files. In addition, we find that most developers do not “modify” the calendar nearly as much as the contacts or invoices file. Therefore we have not found a compelling need to redesign this file using Anchor/Buoy.

In Figures 2.0 – 2.4 you will find a snapshot of the relationships for each of the Core² files. Although this information is available once you purchase the Core² we would like to provide our developers who have not yet purchased the Core² a preview of the relationships before they purchase. Please note that since we continue to enhance the Core, the relationships may be subject to change depending on the version of the Core.

Figure 2.0 -Relationships for Core2 Contacts file

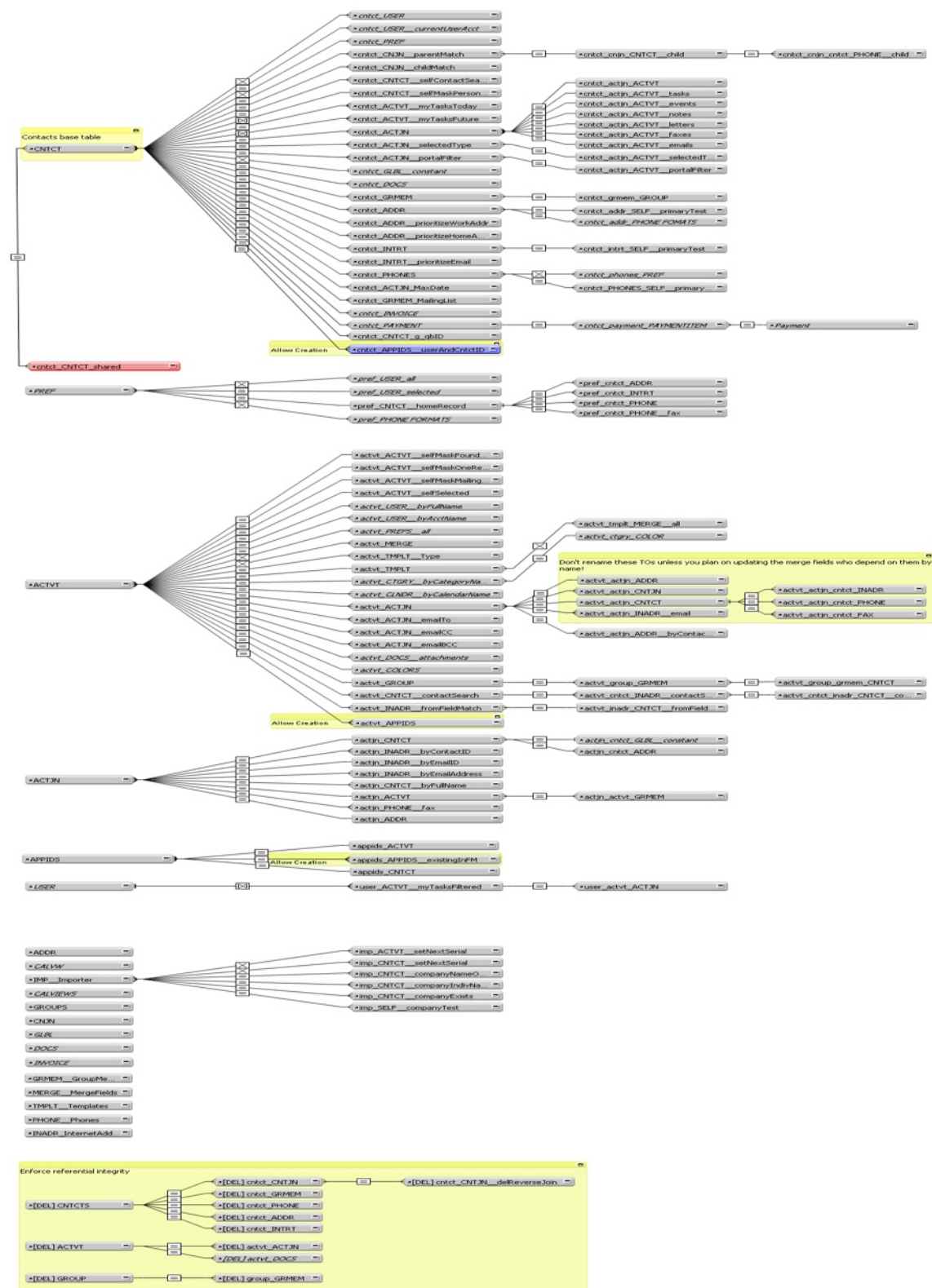


Figure 2.1 - Relationships for Core2 Documents file



- 1- References the documents table. All documents are actually stored in this file
- 2- References the global table in the Preferences file. Used solely for opening initialization script

Figure 2.2 - Relationships for Core2 Preferences file

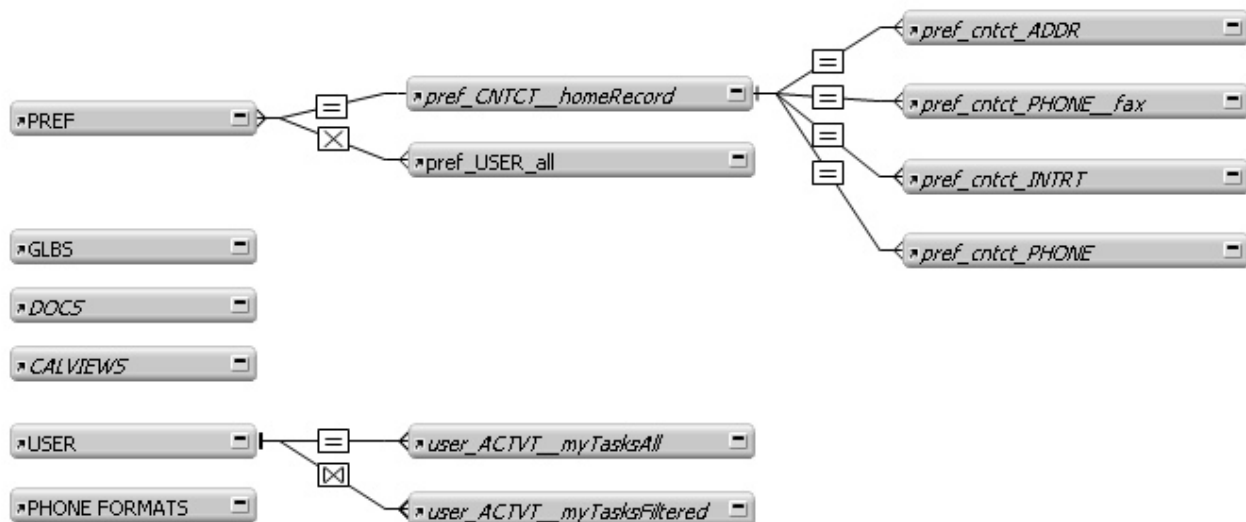


Figure 2.3 - Relationships for Core2 Invoices file

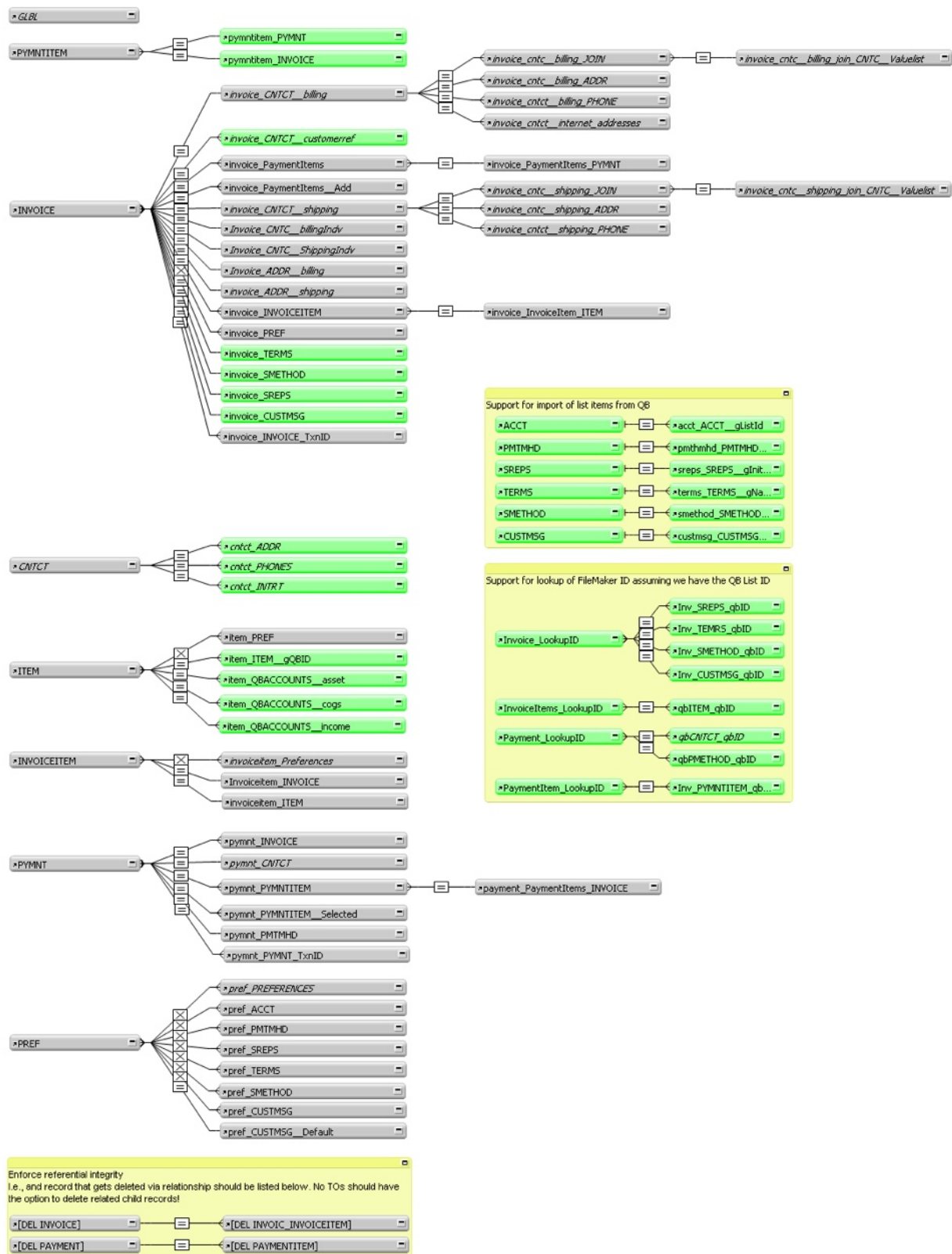
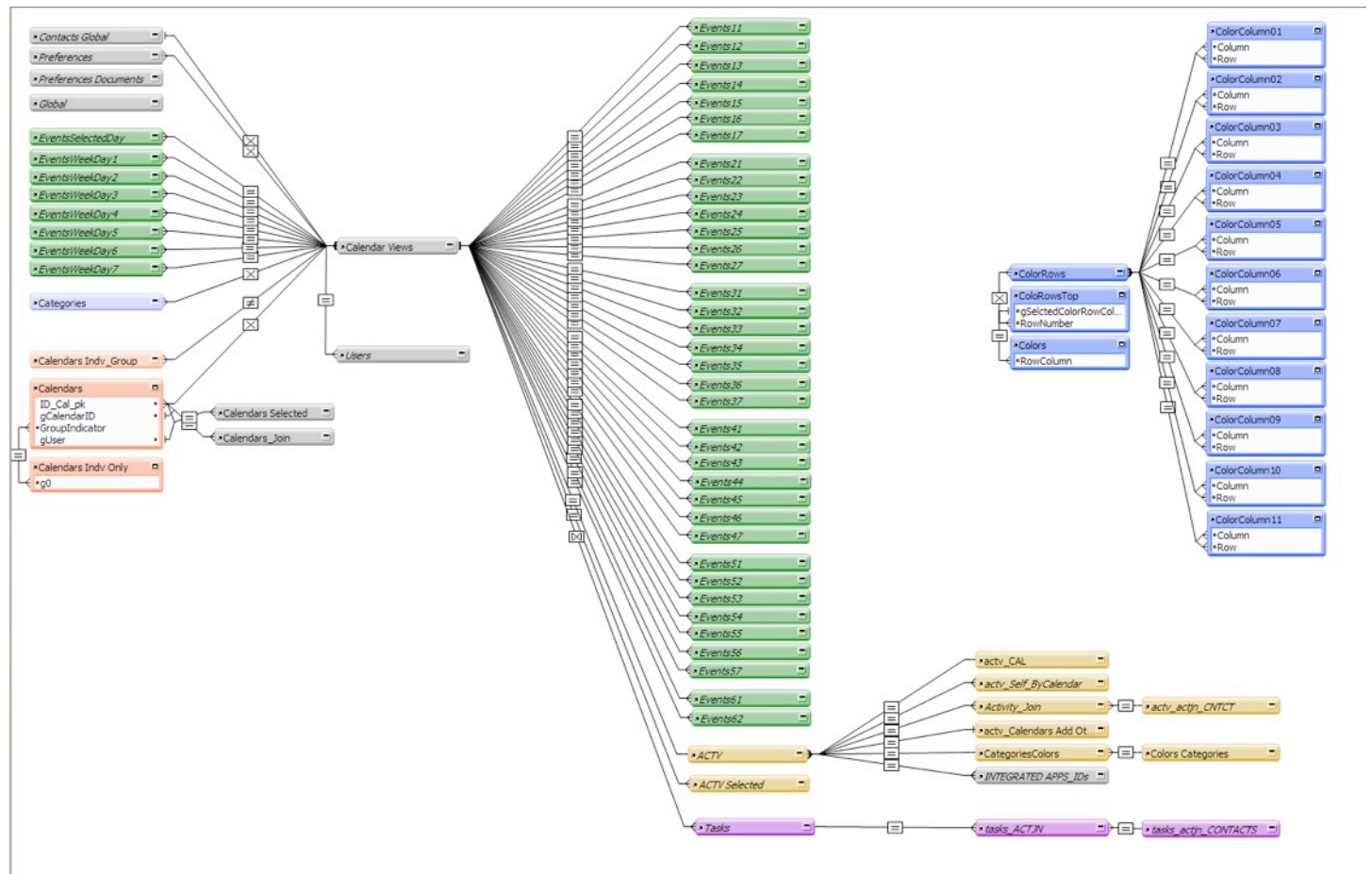


Figure 2.4 - Relationships for Core2 Calendar file



3) Table Descriptions

Since the Core² CRM with Invoices contains over 900 fields and 35 tables, we have chosen to explain each specific table in each Core file including the purpose, exceptions and how the table is linked to other tables.

Figure 3.0 - Core2 Calendar file

| Table | Description/Purpose of Table | Notes/Exceptions | How is the Table linked to any other table? |
|--------------------|--|---|---|
| Calendar_Join | Provides one record for every individual related to a group calendar | Join records will be created only when you associate an individual calendar with a group calendar (or vice versa). | This table is linked directly to the calendars table and is accessed only when a user is in the calendar preferences area. |
| Calendars | Provides one record for every calendar or calendar group | The one record you start out with is for the "General" calendar record. This record exists simple as an initial sample record. | This table provides the drop down list of calendars from all three views (Month, Day, Week). |
| Calendar Views | Provides Month, Week and Day views along with event listing | One record will be created for every unique user's session. This is done largely to avoid record locking. If you had 10 people logging into the calendar, 10 records would be created here. | This table is an interface and direct view into the Activities table found in the Contacts file. |
| Categories | Provides a listing of all categories (user definable). | There are 5 sample category records to get you started. | The categories also contain a link to the colors table to designate a specific color for each category |
| Color Rows | One record for every "row" in the color grid | There is no need to edit this table. The records are simply used for display purposes only. | The color grid appears when designating a category color from the calendar preferences area. This area can be found by way of the preferences button in the main Core Contacts file. |
| Colors | One record for every unique color offered for use on the category record | There is no real need to edit this table. However by creating additional records and specifying their color, more color options could be included. | The colors table links to the colors Rows table to make the presentation of 88 colors appear on a grid. |
| Selected Calendars | One record for every calendar that is currently selected for each user. | This table is more to support the user environment when they select calendars and/or groups to display on the calendar. | This table is linked to the calendars table and is accessed via the month and week views as well as the mechanism when the user "adds to other calendars" while editing a specific event. |

Figure 3.1 - Core2 Contacts file

| Table | Description/Purpose of Table | Notes/Exceptions | How is the Table linked to any other table? |
|------------------|--|---|--|
| Activities | This table holds one record for every email, fax, letter, note, task and event. | When you are creating a mass mail as a campaign, only a single record is created here. Rather, join records are created to represent one record for each recipient. | This table works hand in hand with the activities join table. The contacts table links first to the activities join table, then to the activities table. |
| Activity_Join | This table holds one record for every join created between activities and contacts. | By using a join table, you get the option to link a single task, email, activity etc. to multiple contacts. | Activity Join is a conduit between Contacts and Activities |
| Addresses | This table contains one record for every address entry. | | This table connects directly to the contacts table and allow for a single contact to have virtually unlimited linked addresses. |
| Contact Join | This table holds one record for every join created between linked Contacts. This basically acts as a "self-join" and gives the Core the ability to have a single individual linked to many companies or several companies may have the same linked individual. | | This join table links directly to the contacts table. |
| Contacts | This table holds one record for every contact. The contacts may be an individual or a company. | | The contacts table links to nearly all other tables in one form or another. |
| Group Membership | This table holds one record for every group associated with a particular contact. | Group Membership doesn't hold the group name but rather group name ID. If you have a single contact associated with 4 groups for example, there would be 4 records created in this table. | This table is connected to both the contact table as well as the Groups table. |
| Groups | This table holds one record for every individual group (mailing list group) in the system. | 3 Records here represent the sample data (list A, BE and C). | |

| Contacts File cont. | | | |
|----------------------------|---|--|--|
| Table | Description/Purpose of Table | Notes/Exceptions | How is the Table linked to any other table? |
| Importer | This table holds records that are imported during the "Batch Contact Import" routine found under the FileMaker "File" menu. | This table attempts to take data from a third party application (Excel for example) and "flatten" it in a way that can be imported properly into the Core. | This table is used during the "Batch Contact Import" routine and processed data to be imported into the Contacts, Addresses, Internet Addresses and Phones tables. |
| Internet Addresses | This table contains one record for every address entry. | This table can contain email addresses, web addresses, Skype, AIM and more. | This table connects directly to the contacts table and allow for a single contact to have virtually unlimited linked addresses. |
| Merge Fields | This table contains one record for every sample merge field. | Included are the "main" fields used during a merge. Adding new records to this table and thereby adding available merge fields must be done by someone who has experience in FileMaker relationships. The merge fields here are dependent on knowing the table occurrence in relation to where it is being used. | The templates table links directly to the merge field. This table provides a listing of the "main" merge fields that would typically be used during a mail merge/campaign. |
| IntegratedApps | This table is used to store the unique Outlook and Entourage IDs during the data exchange routines. | This table will populate records only if either the Outlook/Exchange Manipulator or the Entourage Manipulator plug-ins are purchased and integrated | This table links directly to the activities table. It stores 3rd party application IDs when moving data that are tasks or calendar events. |
| Phones | This table contains one record for every phone entry. | | This table connects directly to the contacts table and allow for a single contact to have virtually unlimited linked phone numbers. |
| Templates | This table contains one record for every "template" in the system. Templates are used to store "boilerplate" letters, faxes and Emails and are available when creating letters, faxes and templates as a means to produce commonly used correspondence. | | This table is available to the user under the "preferences button" found on the main screen. This table is linked directly to the activities table where the data is "copied" from the template onto a given letter, fax or email. |

Figure 3.2 - Core2 Documents file

| Table | Description/Purpose of Table | Notes/Exceptions | How is the Table linked to any other table? |
|-----------|--|--|--|
| Documents | This table contains one record for every document in the system. | This table is stored in a separate file to keep this file more agile. It's typical that a "documents" file gets quite large and by separating it from the other components, it makes managing this file easier - especially during the backup process. | The documents table is linked directly to a contact. |

Figure 3.3 - Core2 Preferences file

| Table | Description/Purpose of Table | Notes/Exceptions | How is the Table linked to any other table? |
|---------------|--|--|---|
| Globals | A single record which contains most of the graphics used throughout the entire system. | Each graphic is stored both as a traditional container field as well as a calculated global field. This is designed to specifically allow the editing of a graphic while the file is being hosted using FileMaker server. At the same time the global field is available to all areas o the core without defining a specific relationship. | The Globals table provides graphics for the Contacts file, the calendar file, and the invoice file. |
| Phone Formats | This table contains a single record for every "country phone type" in the system. Based on a given country, the phone number will take on that attribute. | This table applies to those organizations that have contacts outside the country whereby you may want to store a specific phone format by country. We are aware that solution doesn't work in all cases as in many countries multiple phone formats may be used. | This table links directly to the phones table and the addresses table when entering phone numbers, the country is determined and the phone format is applied accordingly. |
| Preferences | This table contains a single record where many of the system-wide preferences are stored. Preferences like the plug-in registration codes, account currently logged in, and global calendar preferences. | DO NOT remove this record. Doing so will break the system. | The preferences table is linked to the Contacts and Calendar file. |
| Users | This table contains one record for every user specifically designed in the preferences area. This file stores items like the user's signature, closing, and specific preferences on how the system interacts with Outlook/Exchange, Entourage etc. | There will be at least one record representing the "admin" user which is setup upon startup. | The users table is used in a variety of areas including but not limited to Letters, Emails, Calendar settings, plug-in data exchange, etc. |

Figure 3.4 - Core2 Invoices file

| Table | Description/Purpose of Table | Notes/Exceptions | How is the Table linked to any other table? |
|---------------|---|--|--|
| Invoices | This table contains one record for every invoice in the system. | | This table links to nearly all other tables in one form or another. |
| InvoiceItems | This table contains one record for every invoice line item in the system. | | |
| Item | This table contains one record for every available item (or product). This table is used to produce the drop down list when adding items to an invoice line item. | This table has many uses and it can be considered a very basic inventory of items available on an invoice. Unlike a true inventory file we don't keep track of quantity "on-hand" or reorder values. | This table is linked mainly to the invoice line items table. |
| Payment | This table contains one record for every payment received. | A Payment is the main transaction (the check received). Not to be confused with a payment item which is how each part of that check applies to a particular invoice. | The payment table is linked mainly to invoices. The payment table is made up of one or more payment line items (splits). |
| Payment Items | This table contains one record for every allocation of a particular payment. If you were to receive a check for \$500 and \$200 went to invoice one and \$300 when to invoice two, you would have 2 records here. | Payment line items are necessary for you to split payments across several invoices. QuickBooks also requires that payment items are separated in their own table from the actual payment. | Payment line items are mainly connected to invoices and their parent table is of course the payment table itself. |
| Preferences | This table contains a single record to keep track of preferences specific to just the invoices file. | Tax Rate and customer message is an example of what kind of preferences are stored in this table. | This table is linked mainly to the invoice table. |

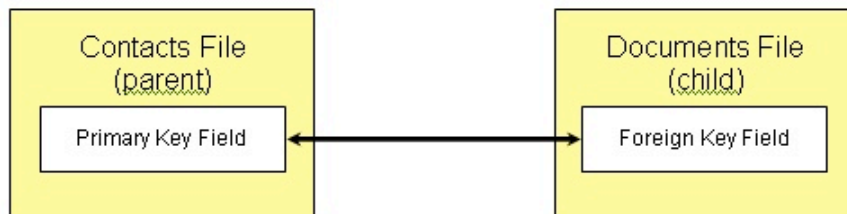
| Invoices File cont. | | | |
|----------------------------|---|--|--|
| Table | Description/Purpose of Table | Notes/Exceptions | How is the Table linked to any other table? |
| qbAccounts | This table contains one record for every account. | This table is mainly used to mirror accounts in the QuickBooks file. This table is necessary if you are creating new items and need to assign those items to a specific account. This is a QuickBooks integration requirement. | This table provides the pull down list of accounts when linking items to accounts. |
| qbCustMsg | This table contains one record for every Customer Message available on the invoice. | QuickBooks keeps all Customer Message records in a separate table and we are required to keep track of the ID for each message. | This account is linked specifically to the invoice table. |
| qbPmtMethod | This table contains one record for every Payment Method available on the invoice. | QuickBooks keeps all Payment Method records in a separate table and we are required to keep track of the ID for each method. Examples include "Check, Visa, MasterCard" | This account is linked specifically to the invoice table. |
| qbSalesReps | This table contains one record for every Sales Rep available on the invoice. | QuickBooks keeps all Sales Rep records in a separate table and we are required to keep track of the ID for each rep. | This account is linked specifically to the invoice table. |
| qbShipMethod | This table contains one record for every Shipping Method available on the invoice. | QuickBooks keeps all Shipping Method records in a separate table and we are required to keep track of the ID for each method. Examples include UPS, FedEx, etc. | This account is linked specifically to the invoice table. |
| qbTerms | This table contains one record for every Term available on the invoice. | QuickBooks keeps all Terms records in a separate table and we are required to keep track of the ID for each Term. Examples include Net 15, Net 30, etc. | This account is linked specifically to the invoice table. |

4) Fields

With over 900 fields we find it more beneficial for the Developer to understand the field prefix and description rather than explain each individual field in detail.

Figure 4.0 – Field Prefix and Descriptions

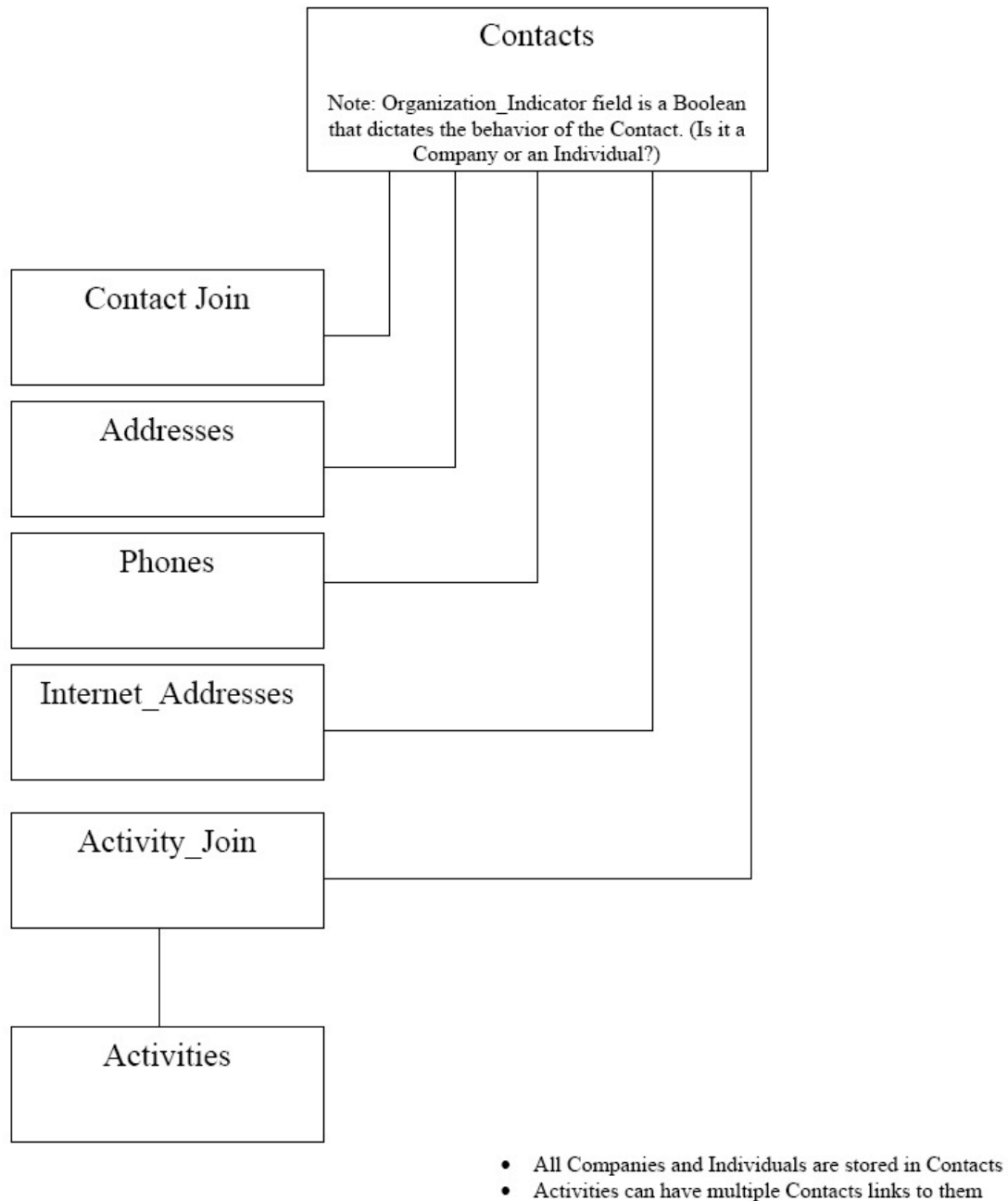
| Field Prefix | Description |
|--------------------------------|---|
| "_k" | Fields that begin with "_k" contain data that is "constant" on every record. They are typically in the form of a calculation. They are created to create an anchor for data used in relationships. Example: <u>_kFax</u> = "Fax" allows us to filter only those phone numbers that have the type "fax" |
| ID | Any field that represents an "ID" or identifier used in a relationship. |
| ID <u>some name</u> <u>pk</u> | Fields ending in "pk" stand for primary key. Fields setup as primary keys contain unique values. These fields are typically used as the main predicate for most "parent" relationships. The primary key typically connects to a child record's foreign key. Primary keys auto populate when new records are added and are always unique. Throughout the core we use a "text" type field for this as it provides the most flexibility. |
| ID <u>some field</u> <u>fk</u> | Fields ending in "fk" stand for foreign key. Fields setup as foreign keys typically contain the ID of the parent record. These fields are typically used as the main predicate for most "parent" relationships. |



5) Entity Relationship Diagram

Here is a basic overview of the main relationships in the Core.

Figure 5.0 – Summarized ERD



6) Scripts

Scripts are organized into groups as follows:

Account Related: Reserved for scripts that are used to either re-log into the file or to add accounts, delete accounts or reset password. These scripts can be triggered from contacts file when you click “add account” in the preferences file.

Activities: Reserved for activity table data management such as creating new emails, letters etc.

Address Book Manipulator: Reserved for scripts related to Apple Address Book integration.

Contacts: Reserved scripts that support functions related to contacts.

Developer: Reserved for scripts that perform developer related/housekeeping activities. This is an area where a single script is used multiple times and called by various scripts.

Entourage Manipulator: Reserved for scripts related to Entourage integration.

FM Books Connector (QuickBooks): Reserved for scripts related to QuickBooks integration.

Global: Not specific to any specific task but rather global in nature such as printing, importing, resizing etc.

iCal Manipulator: Reserved for scripts related to Apple iCal integration.

Invoices: Reserved specifically for invoice and payment related buttons and functions.

Navigation: Reserved for scripts that involve most navigation in or out of the file.

Operations: Reserved for scripts that perform general operations specific to the file.

Outlook/Exchange Manipulator: Reserved for scripts related to Outlook integration.

Preferences: Reserved for management of preferences area in the Core.

Startup/Shutdown: Reserved for scripts that perform when a file opens or closes.

****Other groups not mentioned above may also exist depending on the version of the Core.****

7) TOG Legends

Below you will find the "long name" and/or brief description for the main table occurrence in the system.

Calendar File

Currently not anchor buoy - most Table Occurrences names are spelled out already

Contacts File

CNTCT = Contacts Table

ACTV = Activities Table

ACTJN = Activities Join Table

APPIDS = Integrated Applications IDs Table (for use when storing IDs for Outlook and Entourage mail integration)

ADDR = Address Table

IMP__Importer = Importer table (for use with the batch import process)

GROUPS = Groups Table (to store mailing list groups)

CNJN = Contact Join Table

GRMEM__GroupMembership = Group Membership Table (for store associated IDs with a particular mailing list group)

PHONE = Phones Table

INADR__InternetAddresses = Internet Addresses

Table Occurrences that reference external files

PREF = Preferences Table in the Preferences File

USER = User Table in the Preferences File

CALVW = Calendar Views Table in Calendar File

GLBL = Global Table in Prefs File

DOCS = Documents Table in Documents File

INVOICE = Invoice Table in the Invoices File

GRMEM__GroupMembers

Documents File

Documents = Documents Table

GLBL = Global Table in Prefs File

Preferences = Preferences Table in the Preferences File

Preferences File

PREF = Preferences Table (to store system-wide preferences)

GLBS = Global Table (to store system-wide graphics)

USER = User Table

PHONE FORMATS = Phone Formats table (to store phone formatting by country)

Table Occurrences that reference external files

DOCS = Documents Table in the Documents File

CALVIEWS = Calendar Views Table in the Calendar File

Invoices File

PYMNTITEM = Payment Item Table

INVOICE = Invoices Table

ITEM = Item Table

INVOICEITEM = Invoice Item Table

PYMNT = Payment Table

ACCT = qbAccounts Table

PMTMHD = qbPayment Methods Table

SREPS = qbSales Reps Table

TERMS = qbTerms Table

SMETHOD = qbShip Methods Table

CUSTMSG = Customer Message Table

Table Occurrences that reference external files

GLBL = Global Table in Prefs File

CNTCT = Contacts Table

PREF = Preferences Table in the Preferences File

8) User Accounts and File Security

User Accounts

The Core² CRM uses a dedicated record to maintain the preferences of the Core² CRM for each user. In order for the preferences for each user to work properly, it is necessary to have one corresponding FileMaker account for each user record in each Core² CRM file. This is done for you automatically when you create a new user in the Core² CRM from the preferences area.

After creating a new user account from the preferences area in the Core² CRM, a new FileMaker account is automatically created in all other necessary Core² CRM files. Each new user account is created with the built-in FileMaker "Data Entry Only" privilege set.

If you need to setup additional security options, you can create your own custom privilege set. If you need to add or edit a privilege set or a specific user account privilege, then this needs to be done in EACH Core² CRM file using FileMaker's "Accounts & Privileges." We recommend always using the preferences area in the Core² CRM to create new users unless you are a FileMaker developer or are comfortable with making changes in FileMaker's "Accounts & Privileges" area.

The Core² CRM comes programmed with the following three privilege sets:

Data Entry Only

Full Access

Read-Only Access

The Core² CRM comes with the following two "Full Access" admin accounts:

User = Admin and no password

User = AdminBackup, password = "we do not disclose"

The "AdminBackup" account was created as a back door to assist in the event that you lose your Admin account password. For additional security please change the passwords for these two accounts on ALL Core files and store all passwords in a safe location. Productive Computing assumes no liability or responsibility for lost passwords and issues associated with negligent security set up.

File Security

Throughout the Core² Contacts, Core² Calendar and Core² Prefs files there are "Unlock" and "Reset" scripts. These scripts are worth noting as they might prove beneficial for the developer. For example, the "Unlock" script in the Core² Calendar file also includes a scrip step to allow toolbars. The "Reset" scripts resets the Core back to the factory defaults, which may be useful when installing at a new location.

9) Contact Us

This Developer's Guide was created specifically for you, the Developer. If you feel there are areas that you would like covered in detail, please let us know.

For users of the Core we have created a Quick Start Guide and online video tutorials. These documentation and videos can be found at www.core2crm.com . They provide the end user with basic training and user functionality.

The Core comes fully unlocked allowing you to customize the solution according to your needs. Implementing customizations require knowledge of FileMaker scripting, relationships and calculations. If you need additional support for scripting, customization or setup, then please contact us via the avenues listed below.

Phone: 760-510-1200

Email: support@productivecomputing.com

Forum: www.productivecomputing.com/forum

Please note assisting you with customizations and set up is billable at our standard hourly rate. We bill on a time and materials basis billing only for the time in minutes it takes to assist you. We will be happy to provide you with a free estimate if you fill out a Request For Quote (RFQ) at www.productivecomputing.com/rfq . We are ready to assist and look forward to hearing from you!