

ibase[®] 2.0

Designing Codes

Excerpts from the ibase 2.0 for FileMaker Administrator Manual

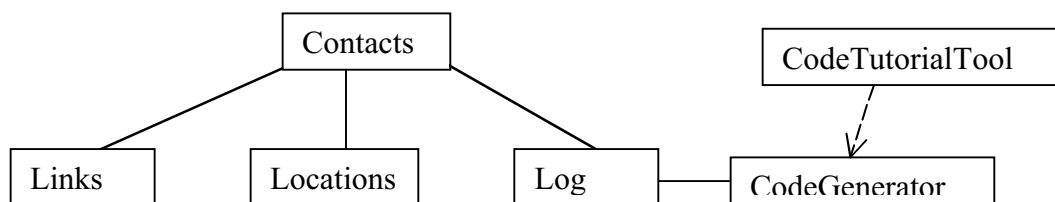
Rev 03/05/2002

4.10: ebase Data Structure: Introduction

Concept:

ebase is a relational database. Instead of storing all data in one file, ebase stores data in many related files linked by an ID number, often the contact ID number. You don't need to track which file you are working in—ebase makes the transition from file to file seamless. But it will help you understand ebase's approach to codes if you have a general idea of how the main files are related.

The following are the main files in the data structure of ebase 2.0 for FileMaker:



For the most part, data stored in Contacts, Locations, and Links describes the "basics" about a contact--the data that has more to do with the contact per se than with the contact's relationship to your organization. Data in the Log file, on the other hand, describes the specifics of a contact's relationship to your organization.

Contacts - is the central hub of ebase 2.0, the starting place for most of the work you'll do. It stores names, profile (demographic) data, and biographical data. There are three types of contact records: individual, family/household and organization.

Locations - stores the multiple addresses you collect for a contact. The ContactLocations file (not shown above) is a connecting file that allows you to share a location among multiple contacts, so when an organization or household moves, you only have to update the address once.

Links - records the links between members of your community. ebase 2.0 enables you to track relationships between contact records, such as between a family/household record and its members; an organization record and its employees; or between two individuals.

Log - records the pieces of data that taken together describe your organization's relationship with a contact: the communications, solicitations, actions taken, availability to do volunteer work, volunteer hours contributed, interests, skills, roles, events attended, and on and on. The range of data that can be stored in the Log is limited only by your imagination.

CodeGenerator - stores the codes needed for organize and retrieving the data stored in the Log file. It's the central supply room, where you go to add new codes to your inventory when you need to record a new type of information. It's where you go to pull a code off the shelf to use in the process of adding a log item to the Log.

CodeTutorialTool - a step-by-step workbook that guides you through the process of creating your own set of set of codes when you first start using ebase 2.0. You then import the code set into the CodeGenerator file and are ready to begin adding data to the Log file.

4.11 Community Mapping and the Code Library: Introduction

Concept:

ebase 2.0's code system is a form of community mapping. The codes you set up, and the log items you create using these codes, capture the details of the relationship between your organization and its communities of constituents. They record the "back and forth" between the organization and its constituents, the common work and mutual support, the shared knowledge—all the things that help create communities.

In the sample code set shipped with ebase 2.0, we've developed codes to map some of the communities that nonprofits typically manage:

- * the community of people who provide financial support (members and/or donors)
- * the community of people who provide support through an investment of time and talent (volunteers and/or activists)
- * the communities of people who help run the organization (board, staff, and committees)
- * the community of people who help with or attend the organization's events.

Not included in the sample code set are some other communities that many nonprofits manage: clients; foundations (grant-tracking), grantees (grant management), vendors and customers (gift shops). And there are others. We welcome contributions of code sets for other communities (and improvements on the existing sample sets). We've set up the ebase 2.0 Code Library on the ebase Web site for sharing code sets within the ebase community.

More complex versions of the sample codes sets shipped with ebase are also posted in the ebase Code Library.

4.12: Codes and Log Items: Introduction

Concept:

ebase 2.0 uses a flexible code system to organize data about a contact's relationship to your organization. Before beginning to use ebase, you'll need to design a set of codes to capture all the data your organization needs to track. Here's an overview of the elements of ebase's code system:

- * **The Log:** stores the pieces of data that taken together describe your organization's relationship with a contact. The Log records all significant activity that happens with a contact, just as a ship's log holds all significant activity on board a ship.
- * **Log Items and Project Steps:** Each item of data that a user enters in the Log is called an "item" or a "log item." A log item usually describes one discrete step in managing a project that involves members of your community. For example, the log items for managing the steps of your membership renewal program might

How to Design Codes in ebase® 2.0 for FileMaker – 3/2/2002

be: 1) Send first renewal notice; 2) Send second renewal notice; 3) Receive membership payment, associate level; 4) Send thank you.

* Add Item: When adding a log item for a contact, a user first picks the item from a list of item names. This list is made up of the titles of the codes that you, as database administrator, have already set up in ebase.

* Codes: ebase's code structure helps you, as database administrator, organize items in advance, so when a user adds an item about a contact it's done in an organized way. Organizing the data with codes makes it easy to retrieve data later when producing a report, sending email, etc. You add the codes when first setting up ebase 2.0 and then periodically thereafter when you need to record new types of information in the Log.

* Codes and Project Steps: Each code should describe a step in a contact's involvement with a program or project you're managing. For example, for an advocacy project, one code describes a contact's interest in the issue, another describes sending an invitation to an event related to the issue, another captures the fact that the contact attended the event, and another describes the follow-up communication you sent to the contact.

* Code Title/Item Name: Each code has a title, a short phrase that describes the project step you are recording. Each code title must be unique. No single word in the code title can exceed 20 characters, and the complete title cannot exceed 60 characters. Your list of code titles serves double-duty as the list of log item names that a user selects from when adding an item for a contact or finding data for a report. The terms "code title" and "item name" can be used interchangeably. For simplicity, we will use the term "code title" in the administrator manual.

* Code Description: Each code also has a code description, a phrase or sentence that amplifies on the code title. In case your code titles aren't self-explanatory, a good code description will help any successor database administrators immeasurably!

* Code Buckets: A code is made up of a maximum of eight subparts or values stored in eight "code buckets." These values strung together make up the code itself, and describe the details of each code and the project step it represents. Not all codes will require values in all eight buckets, but you must use Buckets 1-3 at a minimum. Each code must be unique. Code bucket values must not contain any spaces and no value can exceed 20 characters. The complete code cannot exceed 60 characters, including the seven divider marks between each of the eight values.

Examples of code title and bucket values are:

Code title: "Mbr payment, associate"
Bucket 1: Payment
Bucket 2: Dev (abbreviation for Development)
Bucket 3: Mbr (abbreviation for Member)
Bucket 4: Associate
Complete code: Payment|Dev|Mbr|Associate| | | |

Code title: "Interest in assisting with events"
Bucket 1: Interest
Bucket 2: Admin
Bucket 3: All
Bucket 4: Volunteer
Bucket 5: EventAssist
Complete code: Interest|Admin|All|Volunteer|EventAssist| | | |

As administrator, you set up the code bucket values for each code title. When users choose an item, that item will automatically be categorized with the bucket values you set for that item's code title.

How to Design Codes in ebase® 2.0 for FileMaker – 3/2/2002

* Code Bucket Labels: You build a code by storing values in ebase's eight code buckets. It's best to be consistent about the type of detail you store in each bucket. To help with consistency, ebase is shipped with recommended labels for the code buckets. Absolute consistency is hard to achieve, so be prepared to make a few exceptions to the rules you adopt about which details to store where.

Default (and recommended) code bucket labels are:

Bucket 1: Class
Bucket 2: Owner
Bucket 3: Project
Bucket 4: List/SubProject
Bucket 5: Type/Version
Bucket 6: Method/SubType
Bucket 7: Date/Time
Bucket 8: ShortCode/Legacy

See Help Topic "Code Bucket Labels" for more information on the default labels.

Steps:

1. For more information on how to design your code system, refer to the following Help Topics:

- * Design Your Code System
- * Code Tutorial
- * Code Bucket Labels
- * Classes of Codes (Code Bucket 1)
- * Add/Edit Code

2. Use the Code Tutorial built into ebase 2.0 to help you design your code system. Go to ADMIN-->Code Tutorial.

4.13: Code Tutorial

Concept:

The simplest way to design your codes and set them up initially is to use the Code Tutorial. The Code Tutorial is a workbook designed to help you create your own set of codes before you start using ebase.

The Code Tutorial will ask you questions about your organization and use your responses to create a set of codes that will be used to categorize your data and run reports. At the end of the Tutorial, you will import the codes into ebase's Code Generator.

CAUTION: The Code Tutorial is to be used once and only once--at the very beginning when you set up your codes for the first time. DO NOT use the Code Tutorial to edit codes or to add new codes after you've imported your first set of codes into the Code Generator. Instead, go to ADMIN-->Add/Edit Codes (see Help Topics "Add a Code" and "Edit a Code" for more information.)

Steps:

How to Design Codes in ebase® 2.0 for FileMaker – 3/2/2002

1. Go to ADMIN-->Documentation to open and print the documentation "How to Design Codes.pdf." This .pdf file can be opened with Adobe Acrobat Reader, available for free download at <http://www.adobe.com>. It is important to read this documentation along with using the Code Tutorial.
 2. Go to ADMIN-->Code Tutorial to open the Code Tutorial and click the "Next" button at the bottom of each page to read the orientation provided on pages 1-3 of the Tutorial:
 - * Overview of ebase Log Items, Codes, and Project Steps
 - * ebase Code Structure
 - * Overview of the Code Building Process (steps 4-13 below)
 3. Continue to click the "Next" button at the bottom of each page to design and build your code system (steps 4-13 below).
 4. List your organization's projects and programs. You will use this project list later to "fill" Bucket 3 (Project) for each of your codes.
 5. List the departments or functional areas that are responsible for your projects and programs. You will use this owner list later to "fill" Bucket 2 (Owner).
 6. List the steps (condensed into code titles) involved in managing your projects. Select the department or functional area (owner) responsible for managing each step. Select the project that each step belongs to. At this point you are starting to construct actual codes
 7. List the classes of data you will be tracking, such as payment, action, event, solicitation, etc. You'll use this value list to fill Bucket 1 in the next part.
 8. Assign a class to each code. You are now "filling" Bucket 1 (Class) of each code.
 9. Finish building each code by placing values in the remaining code buckets as needed, to track the desired level of detail and to differentiate between similar codes.
 10. Add a code description for each code title.
 11. Select the pop-up lists on which a code appears on, e.g., solicitation code pop-up list on the payment entry screen.
 12. Select the layouts used when log items are added or viewed.
 13. Review the codes you have created and import the codes into the Code Generator.
-

4.14: Design Your Code System

Concept:

ebase's code system is extremely flexible, almost unlimited in the number and types of codes you can develop. You can develop your codes from scratch, or revise and add to the ebase's sample codes to reflect your own organization's relationships with its contacts.

How to Design Codes in ebase® 2.0 for FileMaker – 3/2/2002

The steps to take in designing your own system of codes (and the steps that the Code Tutorial walks you through) are:

- * List the projects you work on with your communities of constituents.
- * List the departments or functional areas that are responsible for working with communities and projects.
- * List the steps involved in managing your projects and condense these steps into code titles, keeping in mind that these code titles will be the item names users see. Assign owner and project names to each code title.
- * Create classes and assign a class to each of your code titles.
- * Finish building your codes by filling in the other code buckets, as necessary.
- * Add a code description for each code title. This description is not seen by users but it helps administrators remember what each code was designed to describe.
- * Select pop-up lists that the code titles will appear on. For example, you will need to select all current Solicitation codes to appear on the Solicitation pop-up list.
- * Select layouts for viewing and editing. In this step, you determine the data entry screens (and data viewing screens) to use with each class of code.
- * Review your codes for consistency and revise if necessary.
- * Review your draft system of codes with everyone who will use ebase 2.0 and revise if necessary.
- * Import your new codes in ebase's CodeGenerator file.

NOTE: Before beginning to follow the steps outlined in this Help Topic, please read the following Help Topics that provide the background and definitions you will need before beginning to design your codes.

- * "ebase Data Structure: Introduction"
- * "Community Mapping and the Code Library: Introduction"
- * "Codes and Log Items: Introduction"
- * "Code Bucket Labels"
- * "Classes of Codes (Code Bucket 1)"

Steps:

1. List the projects that you work on with your communities.

The best place to start when developing your system of codes is by listing the projects or programs you work on with your various communities. The project names should be general, and not include the specific steps involved in the projects. You will add the steps later in the Code Tutorial.

For example, a membership-based organization might work on a member renewal program with its current members. This general program would be listed below as Renewal. The renewal program includes a series of steps (mailing membership invoices, adding payments, and sending thank yous) that you would list later in the Tutorial (on Page 6). Another project might be member communications, for which steps would be "sending the newsletter" and "regular email updates" (to be listed on Page 6).

Similarly, a land conservation group might work on campaigns or programs such as Smart Growth with its activists, Land Acquisition with its major donors, and Nature Walks with its volunteers and members. Each of these programs would include specific steps to be detailed later in the Code Tutorial.

The communities and projects covered in the ebase sample code sets are:

Community and Projects Table

Community	Projects to Map
Member	Membership Recruitment

How to Design Codes in ebase® 2.0 for FileMaker – 3/2/2002

Community	Projects to Map
	Membership Renewal Campaign
Donor (those who give additional contributions beyond the annual payment considered to be their membership payment)	Donor Recruitment Special Appeals Capital Campaign
Volunteers	Recruitment and Engagement of Volunteers
Activists	Recruitment and Engagement of Activists
Events (presenters and attendees)	Fundraising Party Conference with Pre-Registration
Board	Recruitment and Appointment of Board Members
Staff	Current and Former Staff Lists
Committees	Specific Committees, e.g., Transportation Committee
All (General)	Keeping in Contact (calls, email, letters...) Newsletter and Email List Subscriptions Contact Flags (don't mail, don't solicit...)

2. List the departments or functional areas (owners) that are responsible for working with communities and projects.

List the departments, functional areas, or job titles that are responsible for managing the different projects. (Do not use individuals' names.) The owners of ebase's sample codes are:

- * All (for all departments)
- * Admin
- * Advocacy (for the program areas in an advocacy organization)
- * Dev (for the development/fundraising function)

NOTE: Owner names must not contain any spaces and cannot exceed 20 characters.

3. List the steps involved in managing your projects, condense these steps into code titles, and assign owner and project names to each code title.

List the steps involved in managing each of your projects. For your brainstorming process, use a simple list form, flow chart, or whatever other format works best for you. Then condense the descriptions of your project steps into code titles--brief and clear 4-6 word phrases--and enter the code titles (item names) on page 4 of the Code Tutorial. Each code title must be unique. No single word in the code title can exceed 20 characters, and the complete title cannot exceed 60 characters.

We found it helpful to think first about a process's steps in generic terms. For example, in managing a project such as an annual conference, the generic steps would be: identify, ask, signup, pay, attend, thank. We then listed the specific project steps in fuller and more conversational terms. Then we expressed the step as a code title, choosing the word order that seemed easiest to read and use on an item pick list. Finally, we identified the department responsible for each step. For example:

Project Steps: Annual Conference

Generic Step	Specific Project Step	Code Title	Owner
Identify Interest	Request information on annual conference	Info request, AnnConf 2002	Admin
Invite	Invite to annual conference	Invite to AnnConf 2002	Advocacy
Signup	Sign up to attend annual conference	Registered, AnnConf 2002	Admin

How to Design Codes in ebase® 2.0 for FileMaker – 3/2/2002

Generic Step	Specific Project Step	Code Title	Owner
Pay	Pay registration for annual conference	Event payment, AnnConf	Admin
Attend	Attend for annual conference	Attended AnnConf 2002	Advocacy
Thank	Thank attendees, follow-up with more info	AnnConf thank you, attendees	Advocacy

4. Create classes and assign a class to each of your code titles.

ebase uses classes to group and display item names, to set various flags, and to retrieve data for reports. You will now assign a class to each of the code titles you created in the preceding part.

NOTE: Class names must not contain any spaces and cannot exceed 20 characters.

The classes used for the sample codes are:

- * Action: To record a contact's volunteer work or response to a call to action (e.g., attended hearing)
- * Availability: To record a contact's availability as a volunteer, activist, committee member (day, eve...)
- * Communication: To record communications you send to contacts, (calls to action, newsletters...)
- * Event: To record a contact's actions related to an event you are putting on (RSVP, attendance...)
- * Interest: To record a contact's interests in issues, type of volunteer work, etc.
- * Payment: To record gift donations, dues, purchases, or dollar-equivalents for in-kind gifts.
- * Pledge: To record a contact's pledge to make one or more payments in the future.
- * Pre-ebase: To record a code used in a prior database.
- * Prospect: To record a contact's status as a prospect (for fundraising, advocacy, board membership....)
- * Role: To record the role that someone plays in your organization, e.g., board member, staff.
- * Skill: To record a contact's skill and expertise that your organization may want to draw upon.
- * Solicitation: To record communications for which you want to track the response (payment, actions).

Event is really a subset of Action. Use Event to record constituent's actions that are related to an event you are putting on (RSVP, Signup, Attendance at a party, conference, training, etc.). For communications you send out about the event, use Communication or Solicitation classes (depending on whether you want to link the constituent's response to the solicitation and analyze the effectiveness of different methods of solicitation). Use the Payment class for any payment items related to an event.

Solicitation is really a subset of Communication. Use this class for items for which you want to track the response (be it a donation or an action taken in response to an alert). When you record a payment, you can also record the solicitation that generated the payment from the pop-up list in the Payment For field. You need to do this in order to analyze renewal rates and other response rates.

5. Finish building your codes by filling in the other code buckets, as necessary.

In completing the steps outlined above, you have now filled in code buckets 1-4. You may need to add more information in Buckets 5-8 to differentiate one step (and its code) from another. You don't need to fill all the buckets, but the combined values should capture the detail of the code title and distinguish it from all other codes.

NOTE: Bucket values must not contain any spaces and cannot exceed 20 characters. The complete code cannot exceed 60 characters, including the seven divider marks between each of the eight values.

Here are suggested labels and uses for the ebase code buckets:

Recommended Guidelines for Code Buckets

Bucket # and Label	Questions to Answer when "Filling This Bucket"	Examples
1: Class	What classes of data do you need to track? You MUST use Bucket 1 for Class, but you can develop your own class names.	Action, Availability, Communication, Event, Interest, Payment, Pledge, Pre-ebase, Prospect, Role, Skill, Solicitation
2: Owner	Who is responsible for managing this? You MUST use Bucket 2 for Owner, but can develop your own owner names.	Admin, Dev (for Development), Advocacy, All
3: Project	What are the projects that each owner manages with or for this community?	MbrAcq (for member acquisition); Renewal (for expired members)
4: List or Sub-Project	What group(s) of people does each item involve? What sub-projects are included in this project?	Volunteer, Donor, Attendee, Sponsor, LivingLightly (a payment category), Member (of board or committee); Attorney (for a skill); Rented List#1, AgMethods (as a sub-project of the Food Systems program) etc.
5: Type or Version	What type of activity is involved in this item? What type or version of a solicitation do I need to track	Type of Activity: SignUp, Attend, Canvass, HostParty. Type of Communication: Invoice, ThankYou Version of Solicitation: Invoice, TestA, TestB
6: Sub-Type or Method	What is the method of contact? Note: use "email" in Bucket 6 to take advantage of ebase's inbox feature.	Phone, Letter, Email, PhoneBank, Visit, HouseParty
7: Date or Time	Does a date or time distinguish this item from another?	For volunteer availability: weekend vs. weekday. For mailings: 0402 vs. 0602 For annual events: 2002 vs. 2003
8: ShortCode³ or Legacy	Do you need a short version of the Solicitation codes to print on a solicitation's reply device? (For example to distinguish between two versions of a postcard.) Do you need to store any legacy codes that you import into ebase?	Use whatever coding system you like. The examples given in sample code sets are a combination of the values in the code buckets.

6. Add a code description for each code title.

Add a brief code description for each title to further explain the code by re-phrasing it or adding additional information. You can enter long sentences, even a short paragraph, for the Code Description. This will be displayed in its entirety in the code's record when accessed via ADMIN-->Add/Edit Code. Bear in mind, though, that all other screens will only display the first 5-6 words.

How to Design Codes in ebase® 2.0 for FileMaker – 3/2/2002

Code Description is a required field in ebase, so you need to enter some text here--at a minimum, repeat the code title.

7. Select pop-up lists that the code titles will appear on.

Code titles can appear in several pop-up lists and in the public history (viewable by all users) of a contact's log items:

* Metadata Flag List: You can set up to five "flags" to appear in the metadata bar on the right side of the Contacts Overview screen, e.g., "Board Member - Current" or "Board Member - Former." Selecting "YES" will include the code on the pop-up list in ADMIN-->Customize Metadata, so that you can select the code as one to use when setting up the metadata flag.

* Solicitation List: When entering a payment, you can also enter the solicitation code that generated that payment, in order to later analyze a solicitation's response rate. Selecting "YES" will include the code in the "Solicitation Code" pop-up list on the Payment screen. NOTE: You can later remove old solicitation codes from the pop-up list by going to ADMIN-->Add/Edit Codes, selecting the relevant code, and clicking in the "See in Solicitation List" field to deselect it.

* Pledge List: When entering payments on a pledge, you will need to enter the pledge code that the payment is linked to. Selecting "YES" will include the code in the "Pledge" pop-up list to be used when entering a pledge payment.

* Public History: This option determines whether log items linked to this code appear on the public history of log items for a contact. We recommend selecting "YES" for all codes except those in the Payment and Pledge classes.

* Member/Recurring Payment: This option determines whether the code is considered a membership payment (or a recurring payment that is like a membership payment) and is used to determine currency of membership. It is used to set the Current Member flag in the metadata bar on the right side of the Contacts Overview screen.

8. Select layouts for viewing and editing.

You're almost done--there are only a few last bits of data to complete for each code, and these can be entered quite efficiently in the Code Tutorial.

ebase has specialized layouts for entering data in log items and viewing log items in each code class. Use the guide below to select an edit and view layout for each of your codes:

Layout Selection Guidelines

Class	"Edit with" Layout	"View with" Layout
Action	Note Edit; 9	Note Edit; 9
Availability	Note Edit; 9	Note Edit; 9
Communication	Communication Edit; 12	Communication Edit; 12
Event	Note Edit; 9	Note Edit; 9
Interest	Note Edit; 9	Note Edit; 9
Payment (purchase)	Purchase Edit; 11	Payment View; 2
Payment (membership; gift)	Payment Edit; 1	Payment View; 2
Payment (on a pledge)	Payment Pledge Edit; 4	Payment Pledge View; 3
Pledge	Pledge Edit; 6	Pledge View; 7
Prospect	Note Edit; 9	Note Edit; 9
Role	Note Edit; 9	Note Edit; 9
Skill	Note Edit; 9	Note Edit; 9

Class	"Edit with" Layout	"View with" Layout
Solicitation	Communication Edit; 12	Communication Edit; 12

The numbers after the codes are used for technical reasons, required by FileMaker for calling up layouts.

9. Review your codes for uniqueness and consistency and revise if necessary.

Make sure that all code titles are unique. You'll run into big problems with data accuracy if they are not! You may need to go back and make changes after you've reviewed the list. It's much easier if you spot any duplications or inconsistencies now, before users begin to add items to the Log.

Even though ebase 2.0 doesn't require a much consistency in the type of data stored in the code buckets, its good to strive for consistency anyway because the reports shipped with ebase 2.0 (or those you create later for yourself) will rely on certain categories of data being stored in certain buckets. Follow the general format set out by the code bucket labels to ensure that the reports included in ebase function properly.

10. Review your draft system of codes with everyone who will use ebase and revise if necessary.

ebase codes need to reflect the information tracking and reporting needs of everyone who will use ebase®. You undoubtedly consulted other users when developing the codes, but take time to do a final code review with all who will use ebase or the reports it generates.

11. Import your new codes to ebase (using the "Import" button on the last page of the Code Tutorial).

4.20: Codes Bucket Labels

Concept:

A code is made up of a maximum of eight subparts or values stored in eight "code buckets." These values strung together make up the code itself, and describe the details of each code and the project step it represents.

It's best to be consistent about the type of detail you store in each bucket. To help with consistency, ebase is shipped with recommended labels for the code buckets. Absolute consistency is hard to achieve, so be prepared to make a few exceptions to the rules you adopt about which details to store where.

NOTE: You can customize the code bucket labels when you set up your own code system if the suggested titles don't meet your needs, but please note that if you change the type of coding in the buckets, some reports shipped with ebase may not function properly.

The default (and recommended) code bucket labels are:

- * Bucket 1: Class – what classes of data do you need to track for your projects (solicitations, payments, interests, skills, actions, etc.)?
- * Bucket 2: Owner – who is responsible for managing each project? which department or functional area?
- * Bucket 3: Project - what projects do you need to track in ebase®?

How to Design Codes in ebase® 2.0 for FileMaker – 3/2/2002

- * Bucket 4: List/SubProject : what group(s) of people does each project involve? or what subprojects do you need to track?
 - * Bucket 5 - Type/Version: what type of activity is involved in this step? or what type or version of solicitation do you need to track?
 - * Bucket 6 - Method/SubType: what is the method of contact? or what different subtypes (or other details) do you need to track. NOTE: Be sure to use this bucket to indicate "email" for email communications in order to take advantage of ebase's email inbox processing.
 - * Bucket 7 - Date/Time: what date or time distinguishes this code from another? (Sep02 vs. Oct02)
 - * Bucket 8 - ShortCode/Legacy: do you need a short version of a solicitation code to print on a reply device (must be a unique value)? do you need to store legacy codes?
-

4.21: Classes of Codes (Code Bucket 1)

Concept:

Each code in ebase is built using values stored in up to eight sub-parts called "code buckets." The first code bucket stores information about a code's class, the major category to which the code belongs. ebase ships sample code classes for Bucket 1. You, as database administrator, can customize the class list.

The class values shipped with ebase are:

- * Action
- * Availability
- * Communication
- * Event
- * Interest
- * Origin
- * Payment
- * Pledge
- * Pre-ebase
- * Role
- * Skill
- * Solicitation

Here's how one activist organizer who uses ebase explains how to use the sample code classes:

* Action - As much as we serve our constituencies, our constituencies also serve us. By volunteering, writing decision-makers, hosting house parties and more, your constituents take many actions on your behalf. The Action class is for that kind of information.

* Availability –Use the Availability class to record when someone is available to help your organization by volunteering, attending committee meetings, attending public hearings, etc. When you need to schedule meetings and especially when you need to target a request for assistance quickly, it helps to know in advance when a constituent is available.

How to Design Codes in ebase® 2.0 for FileMaker – 3/2/2002

* Communication - Use the Communications class for recording what communications you send to the people in your database--phone calls, emails, letters, education and outreach mailings, action alerts, formal or chance meetings. Note: Use the Solicitation class for those communications intended to solicit payments or actions if you want to be able to track the response to those communications. (see Solicitation below)

* Event - When someone RSVPs or attends an event or group meeting, use the Event class to record the information.

* Interest - One of the best ways to keep our volunteers and members happy is to give them what they want. In other words, we need to tell them about the issues they're interested in and give them the opportunity to take action in ways they like. Use the Interest class to record that kind of information, so you can easily find, for example, everyone interested in air quality, or everyone who wants to work on a Get Out the Vote project.

* Payment - The Payment class captures information about any kind of one-time payment you receive, whether it's membership dues, major gifts, foundation grants, or merchandise sales. Anytime you get a check, use this class to record the information.

* Pledge - Recurring, ongoing donations and membership payments get recorded here. For example, use the Pledge class to capture a credit card pledge of \$25 per month for 12 months made during your spring pledge event.

* Pre-ebase – You may have a lot of old codes and data related to those codes from your former database that you want to keep track of but not work with actively. Park those codes in this class.

* Prospect – All kinds of prospective relationships get recorded in the Prospect class: major donor prospects, prospective board members, membership prospects, prospective clients, prospective funders. Someone who plays an active role with your organization wearing one hat could also be classified as a prospect for some other type of relationship with your organization.

* Role - What are the roles that people play in relation to your organization? Use the Role class to record information about memberships, staff, board members, prospective donors, etc. Codes in this class are also what you would use to capture information about roles people play outside of your organization, but are relevant to your work - state legislators, members of the press, etc.

* Skill - What skill does this contact have that might be important to your organization? An arts organization would use code titles in the skill class to record the type of work an artist does (modern dance, sculpture, chamber music, etc.).

* Solicitation - The Solicitation class is for recording the communications intended to elicit payments from contact, such as membership invoices, direct mail to rented lists for membership prospecting, special appeals, etc. Solicitation code titles will appear in the list of items a user selects from to link a payment back to the solicitation that generated it.

Steps:

1. Examine the code classes shipped with ebase and decide whether you need to add or subtract any classes when you set up your code system.
2. See Help Topics "Code Tutorial" to evaluate whether you have identified all the code classes needed for organizing the information users will want to log about your contacts.