

**AccountMate 7 for SQL/Express**

**System**

**Component Reference Guide**

**“Sample”**

Copyright © 1997 - 2006 by

AccountMate Software Corporation

88 Rowland Way

Novato, CA 94945

United States of America

[www.accountmate.com](http://www.accountmate.com)

# Copyright

---

## **COPYRIGHT WARNING**

All rights reserved. This publication and the related software and documentation are protected by the copyright laws of the United States and other countries. No part of this publication and the related software and documentation may be reproduced in any form or by any means without the written permission of AccountMate Software Corporation. The copying, duplication, sale or other unauthorized use or distribution of this material shall constitute a violation of such copyright laws, resulting in civil damages of up to \$100,000 per violation, plus criminal penalties of up to one year imprisonment and/or a \$25,000 fine.

## **GENERAL DISCLAIMER**

Although AccountMate Software Corporation has made reasonable efforts to provide accurate, reliable information on this documentation, AccountMate Software Corporation assumes no responsibility for the accuracy of the information. Therefore, all information and materials are provided "AS IS" without warranty of any kind. ACCOUNTMATE SOFTWARE CORPORATION DISCLAIMS ANY WARRANTIES, EXPRESS OR IMPLIED, REGARDING ANY MATTER WHATSOEVER RELATING TO OR REFERENCED BY THIS DOCUMENTATION, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE. ACCOUNTMATE SOFTWARE CORPORATION MAKES NO WARRANTY THAT THE DOCUMENTATION WILL MEET YOUR REQUIREMENTS, OR THAT THE RESULTS OBTAINED FROM USING THE DOCUMENTATION OR FROM ANY SERVICES OFFERED THROUGH THE DOCUMENTATION WILL BE ACCURATE OR RELIABLE. AccountMate Software Corporation reserves the right to modify the product and service information contained in this documentation without notice to current or prospective customers.

## **LIMITATION OF LIABILITIES**

ACCOUNTMATE SOFTWARE CORPORATION SHALL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR REVENUES, COSTS OF REPLACEMENT GOODS, LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS ANY AND ALL SECTIONS OF THIS DOCUMENTATION OR ANY ACCOUNTMATE SOFTWARE CORPORATION PRODUCT, DAMAGES RESULTING FROM USE OF OR RELIANCE ON THE INFORMATION PRESENT, EVEN IF ACCOUNTMATE SOFTWARE CORPORATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY. THEREFORE, SOME OF THE ABOVE LIMITATIONS MAY NOT APPLY.

## **TRADEMARKS**

AccountMate® is a registered trademark of AccountMate Software Corporation

August 2006

# Contents

---

<b>Introduction.....</b>	<b>5</b>
<b>1 - AccountMate Program Components .....</b>	<b>6</b>
Files .....	6
Data .....	7
Documents .....	9
Class Libraries.....	10
Code.....	10
Others .....	11
<b>2 - Class Libraries .....</b>	<b>12</b>
Non-Visual Class Libraries.....	12
Visual Class Libraries.....	28
<b>3 - System Manager Functions and Procedures.....</b>	<b>97</b>
AddDay().....	97
AddOnModList() .....	97
AddOnSourceList() .....	98
AllDigits() .....	98
AmInLock() .....	98
AmLock().....	99
AmMsg().....	99
AmPm().....	100
AmTran().....	100
AmTranMemo().....	100
BeginSqlTrans() .....	101
BlankSqlDate().....	101
CloseCursor() .....	101
CtlKeyCaption().....	102
DisableViewMemoUpdate().....	102
DocNoExisted().....	102
DToYMD() .....	103
EmptyToNull() .....	103
EndSqlTrans().....	103
GetConnect().....	104
GetDelNo().....	104
GetFullName() .....	104
GetKeyCaption().....	105
GetMonthDiff() .....	105
GetNewDocNo().....	105
GetNewSeq().....	106
GetNewUID().....	106
GetRecurCycleCount().....	106
GetSqlData().....	107
GetValue().....	107
InitEnvironment .....	107
InitSessionObject().....	108
InitViewParms().....	108
IsTrue() .....	108
LogicalToInt().....	108
NullToEmpty() .....	109
OpenExcl().....	109
PrepareSql().....	109

<i>RefreshSqlView()</i> .....	110
<i>ReleaseSessionObject()</i> .....	110
<i>SetSqlData()</i> .....	110
<i>SortBySeq()</i> .....	111
<i>sp_AssignUID()</i> .....	111
<i>SqlDtoS()</i> .....	111
<i>TrimKey()</i> .....	112
<i>TToYMD()</i> .....	112
<i>UnprepareSql()</i> .....	112
<i>ViewTxtFile</i> .....	113
<b>4 - SQL Pass-Through String Functions.....</b>	<b>114</b>
<i>odbcDate()</i> .....	114
<i>odbcDateTime()</i> .....	114
<i>odbcIfNull()</i> .....	115
<i>odbcLogical()</i> .....	115
<i>odbcSpace()</i> .....	115
<i>odbcString()</i> .....	116
<b>5 - Program Flow .....</b>	<b>117</b>
<i>Loading AccountMate</i> .....	117
<i>Selecting a Menu Option</i> .....	117
<b>6 - Programming Style.....</b>	<b>118</b>
<i>Naming Convention</i> .....	119
<i>User Interface</i> .....	126
<i>Coding Standards</i> .....	129

# Introduction

---

This manual is designed to give readers an overview of the System Manager functions, procedures, objects, visual class libraries, and other system information used to build the AccountMate application.

The *AccountMate Program Components* chapter describes all the different file types used to build AccountMate. It details the various sub-directories that are created during the installation and the types of files that are installed in each sub-directory.

The *Class Libraries* chapter lists both the non-visual and visual class libraries that are used in the system. Explanation of each class property and method is included in this chapter.

The *System Manager Functions and Procedures* chapter contains a list of available functions and procedures. Parameters and return data type are also described in the chapter.

The *SQL Pass-Through String Functions* chapter contains a list of VFP-like functions that are used to translate values into strings used to build SQL statements.

The *Program Flow* chapter provides a brief overview on the internal works of AccountMate. For example, it includes the logic flow during the system loading.

The *Programming Style* describes the coding standards used. This chapter will help AccountMate developers understand the program. It is also recommended for developers who like to maintain coding and interface consistent to that of AccountMate.

# 1 - AccountMate Program Components

## Files

AccountMate 7 for SQL/Express is composed of many different types of files. The system is made up of databases (i.e., remote views, connections, etc.), free tables, forms, reports, labels, class libraries, programs, menus, text files and other files.

When the software is completely installed with the Source Code, you will see the following directories and files under the AccountMate program directory (e.g., C:\AMSQL). The directories and files that appear in **bold** on the following table denote that they are installed from the Source installations.

Level 1 Directories / File Types	Level 2 Directories / File Types	Level 3 Directories / File Types
*.EXE, *.DBF/CDX/FPT <sup>1</sup> , *.H, *.VCX/VCT <sup>1</sup> , *.CHM, CONFIG.FPW, *.FRX/FRT, other driver files		
\Attachments		
<b>\Bitmap</b>	<b>*.BMP, *.ICO, *.CUR</b> <b>\VisualClass</b>	<b>*.BMP</b>
\Crystal	\AP \AR \BR \CO \GL \IC \Labels \MI \PO \PR \RA \SO \template	*.PRG/FXP, *.RPT *.PRG/FXP, *.RPT *.PRG/FXP, *.RPT *.PRG/FXP, *.RPT *.PRG/FXP, *.RPT *.PRG/FXP, *.RPT *.PRG/FXP, *.RPT *.PRG/FXP, *.RPT *.PRG/FXP, *.RPT *.PRG/FXP, *.RPT *.PRG/FXP, *.RPT *.PRG/FXP, *.RPT *.PRG/FXP, *.RPT *.PRG/FXP, *.RPT Landscape.RPT, Portrait.RPT
\Data	*.DBC/DCT/DCX	
\Documentation	\Getting Started Guide *.PDF	*.EXE
\ErrorLog		
<b>\Forms</b>		
\Reports	\AP \Custom \GL \PR	Apchck??*.FRX/FRT *.FRX/FRT Prchck??*.FRX/FRT
\Rptmod	README.TXT	
<b>\Source</b>	<b>*.SCX/SCT, *.PRG,</b> <b>README.TXT</b>	

## 2 - Class Libraries

---

The classes are the major component in object-oriented programming. AccountMate defines numerous visual and non-visual classes for use building the system. There are 5 visual class libraries, namely: Amsqlbsc.VCX, Amsqlftp.VCX, Amsqledt.VCX, Amsqlrpt.VCX and Amsqltax.VCX. Non-visual class libraries are defined in the System Manager and are available for all the modules to use.

### Non-Visual Class Libraries

The non-visual class libraries contain definitions for CUSTOM classes. These classes are created as global objects so that their properties and methods can be referenced or called throughout the system. It is essential to know and understand the uses of the properties and methods of each non-visual class. The system manager classes cannot be modified, only the module classes come with source code. The following is a list of properties for the system manager non-visual classes used throughout the system.

Property	Description
oSQLCompany.cCompanyID	Company ID
oSQLCompany.cIDName	Company ID name
oSQLCompany.cCompanyName	Company name
oSQLCompany.cCntryTax	Company Country Tax Code
oSQLCompany.cCntry	Company Country
oSQLCompany.cAddr1	Company address (address line 1)
oSQLCompany.cAddr2	Company address (address line 2)
oSQLCompany.cCity	Company address (city)
oSQLCompany.cDbBuildNo	Company database build #
oSQLCompany.cErrLogDir	Company error log directory
oSQLCompany.cState	Company address (state)
oSQLCompany.cZip	Company address (zip code)
oSQLCompany.cCountry	Company address (country)
oSQLCompany.cPhone	Company phone #
oSQLCompany.cFax	Company fax #
oSQLCompany.cLblTax1	Company Tax Label 1
oSQLCompany.cLblTax2	Company Tax Label 2
oSQLCompany.cLblTax3	Company Tax Label 3
oSQLCompany.cLblTax4	Company Tax Label 4
oSQLCompany.cLblTax5	Company Tax Label 5
oSQLCompany.cTaxFld1	Company Tax Field 1
oSQLCompany.cTaxFld2	Company Tax Field 2
oSQLCompany.cTaxFld3	Company Tax Field 3
oSQLCompany.cTaxFld4	Company Tax Field 4
oSQLCompany.cTaxFld5	Company Tax Field 5
oSQLCompany.cTaxMask1	Company Tax Mask 1
oSQLCompany.cTaxMask2	Company Tax Mask 2
oSQLCompany.cTaxMask3	Company Tax Mask 3
oSQLCompany.cTaxMask4	Company Tax Mask 4
oSQLCompany.cTaxMask5	Company Tax Mask 5

## 3 - System Manager Functions and Procedures

---

This section contains a list of System Manager functions and procedures used throughout the system. It is important to understand these procedures and functions and determine when each procedure or function should be used.

### AddDay()

This function adds a number of days to a date field, based on a weekly, monthly, bimonthly, quarterly, semi-annual, or annual schedule. This function is typically used to compute the next recurring date when generating recurring transactions.

Parameter	Description
tdOld	The date to which the number of days will be added
tcAddBy	Indicates whether to add dates according to a weekly, monthly, bimonthly, quarterly, semi-annual, or annual schedule “W” — weekly “M” — monthly “B” — bimonthly “Q” — quarterly “S” — semi-annually “A” — annual
tlLastDay	Indicates whether the last day of the month should be used (optional) Default: .F.
Return Type	Description
Date	New date
Example	replace dNextInv with AddDay(Arrcri.dNextInv, Arrcri.cRecurCycl, Arrcri.lLastDay) in Arrcri

### AddOnModList()

This function populates the list of forms that will be executed using the source file (e.g., file in the FORMS folder) excluding AccountMate forms.

Parameter	Description
tcFileName	Filename
Return Type	Description
<none>	
Example	=AddonModList('FORM1.SCX')

## 6 - Programming Style

---

AccountMate has been developing the Visual AccountMate and AccountMate accounting system since 1984. Over the years, we found there are two important concerns in designing the software – user interface and programming style. The AccountMate systems are designed with a set of conventions and standards. The development team follows four basic philosophies during any AccountMate product development. They are user-friendliness, ease of daily operation, interface and coding consistency, and ease of modification.

These philosophies uphold the AccountMate tradition of not only giving users the ease of use but also the ease of modifying the source code in order for the program to better suit their special needs.

User-friendliness is very important in the design of a complex computer software program. This allows the users to comprehend the software with minimal training. Providing clear and concise windows, messages and functions creates a user-friendly system.

By presenting the accounting functions in the order that mimics the daily operations of a business, providing on-line help function and on-the-fly data validation, it makes the program easier to use and minimizes the human errors in the program. Drill-down capability to view detail information or access another function will increase the users' performance. Ability to access functions, in the same module or maybe even another module, without exiting the current function, such as through the Shortcut Pane, also increases a company's productivity.

Any part of AccountMate system is designed under a set of programming standards – both internally and externally. This allows a user to master all the modules without learning all of them. For consultants or programmers, they do not have to read the entire program coding before they start a modification. The more consistent the software is, the easier it is to use and to customize.

Since every business is run differently, their accounting functions can also be different. For example, a company may run a report daily, but another company may never look at that report or the reporting data need to be reorganized before it is useful for analysis. The release of source code allows the consultants to tailor the software to their clients' needs. The programming conventions and standards used make the AccountMate systems modifiable.

To successfully modify the software and maintain the source code, a consultant must define and implement all possible naming conventions and standards. The rest of the chapter will discuss the programming styles used in AccountMate.

## Naming Convention

Most of the following naming convention follows the industry standard. Some of them can also be found in the Visual FoxPro user manual. The industry standards are used to minimize the learning path for consultants.

### Database

Each company in AccountMate has one database (.DBC). The database contains all the remote views that are used by the module functions. The System Manager also has one database called AMWSYS.DBC, which contains the remote views that are used by the system manager. It is important to note that the System Manager have both back-end and front-end tables. Refer to the *Database Reference Guide* for the list of AccountMate tables.

### Back End Tables

Back end table names have 6 characters, except for history and archived tables, which are 7 characters long. The first two characters of the table name is the module ID where the table is primarily used. Some tables have a prefix of “CO”, and they are used by most of the modules. (e.g., Cobank) A four-character table description follows the module ID. History and archived tables have an “H” as the seventh character.

<b>a</b>	<b>r</b>	<b>i</b>	<b>n</b>	<b>v</b>	<b>c</b>	<b>(h)</b>
Module ID		4-letter Table Description				History

Position	Description
1 – 2	Module ID
3 – 6	Abbreviation of the table description
7	An “h” to indicate history or archived tables

Here are a few examples:

Table Name	Description
Arcust	AR Customer Table
Apvend	AP Vendor Table
Arcusth	AR History Customer Table
Apvendh	AP Archived Vendor Table

### Front End Tables

Front end tables are free tables (.DBF). These free tables are prefixed with “AM” and are located in the AccountMate program directory (e.g., C:\AMSQL). Their table names vary from 5 to 8 characters long.

### Data Fields

The data field names vary from 3 to 10 characters, with the first character denoting the data type of the field.

<b>c</b>	<b>c</b>	<b>u</b>	<b>s</b>	<b>t</b>	<b>n</b>	<b>o</b>			
Data	Abbreviated field description								