

PipeLine 5.1

Technical Documentation

PipeLine 5.0

Technical Documentation

USAID | DELIVER PROJECT, Task Order 1

The USAID | DELIVER PROJECT, Task Order 1, is funded by the U.S. Agency for International Development under contract no. GPO-I-01-06-00007-00, beginning September 29, 2006. Task Order 1 is implemented by John Snow, Inc., in collaboration with PATH; Crown Agents Consultancy, Inc.; Abt Associates; Fuel Logistics Group (Pty) Ltd.; UPS Supply Chain Solutions; The Manoff Group; and 3i Infotech. The project improves essential health commodity supply chains by strengthening logistics management information systems, streamlining distribution systems, identifying financial resources for procurement and supply chain operations, and enhancing forecasting and procurement planning. The project also encourages policymakers and donors to support logistics as a critical factor in the overall success of their health care mandates.

Recommended Citation

Blankenship, Lisa. 2011. *PipeLine 5.0: Technical Documentation.* Arlington, Va.: USAID | DELIVER PROJECT, Task Order 1.

Abstract

Develop technical documentation for PipeLine so that the application is well understood by technical persons, easily maintainable and transferrable. This will ensure that this application’s software development life cycle (SDLC) be managed effectively in future.

Cover Photo: Woman enters shipment data in Zimbabwe, on November 16, 2009.

**USAID | DELIVER PROJECT**

John Snow, Inc.

1616 Fort Myer Drive, 11th Floor

Arlington, VA 22209 USA

Phone: 703-528-7474

Fax: 703-528-7480

Email: ask[deliver@jsi.com](mailto:deliver@jsi.com)

Internet: [deliver.jsi.com](file:///C:\Users\lblanken\Documents%20and%20Settings\plancaster\Local%20Settings\Temp\XPgrpwise\www.deliver.jsi.com)

Contents

[Acronyms xvii](#_Toc293914878)

[Acknowledgments xix](#_Toc293914879)

[Introduction 21](#_Toc293914880)

[Overview 21](#_Toc293914881)

[What PipeLine Can Do for You 21](#_Toc293914882)

[Why Use PipeLine? 22](#_Toc293914883)

[Who Should Use PipeLine? 22](#_Toc293914884)

[PipeLine Software Functions 23](#_Toc293914885)

[Pipeline Monitoring 23](#_Toc293914886)

[Procurement Planning 23](#_Toc293914887)

[Technical Architecture 24](#_Toc293914888)

[Process Flows 25](#_Toc293914889)

[Appendices 25](#_Toc293914890)

[The Reddick VBA (RVBA) Naming Conventions, Version 6.01 25](#_Toc293914891)

[System Requirements 26](#_Toc293914892)

[Installation and Configuration 27](#_Toc293914893)

[Installation Instructions 27](#_Toc293914894)

[How to Install PipeLine 27](#_Toc293914895)

[Before You Begin 27](#_Toc293914896)

[Installing PipeLine from a CD 27](#_Toc293914897)

[Installing PipeLine from the Internet 28](#_Toc293914898)

[How to Start PipeLine 28](#_Toc293914899)

[Starting PipeLine from the Windows desktop 28](#_Toc293914900)

[Starting PipeLine from the Windows taskbar 28](#_Toc293914901)

[Reinstalling PipeLine 28](#_Toc293914902)

[Converting Your Existing Data Files 29](#_Toc293914903)

[Run-time installation 30](#_Toc293914904)

[Development environment installation 32](#_Toc293914905)

[Build process 33](#_Toc293914906)

[Installer 33](#_Toc293914907)

[Mechanics for Creating the CD Installer 33](#_Toc293914908)

[Create files with SageKey 33](#_Toc293914909)

[Enter Application Information 33](#_Toc293914910)

[Select Application Files 34](#_Toc293914911)

[Front-end Application 35](#_Toc293914912)

[Back-end Database 35](#_Toc293914913)

[Enter Extra Information 35](#_Toc293914914)

[Select Your Icon 35](#_Toc293914915)

[Select Shortcuts 35](#_Toc293914916)

[Store the StartAccess Command Line in the registry & System.mdw 35](#_Toc293914917)

[Custom Help File 35](#_Toc293914918)

[Select Runtime Files 35](#_Toc293914919)

[Access Runtime Files: 36](#_Toc293914920)

[Template File: 36](#_Toc293914921)

[End User License Agreement file: 36](#_Toc293914922)

[Select Other Files 36](#_Toc293914923)

[File Details 37](#_Toc293914924)

[Re-Scan 38](#_Toc293914925)

[Enter Registry Keys 38](#_Toc293914926)

[Design Welcome Dialog 39](#_Toc293914927)

[Select Merge Modules 39](#_Toc293914928)

[Enter Build Options 40](#_Toc293914929)

[Build Location: 41](#_Toc293914930)

[Install File Name 41](#_Toc293914931)

[Installation Options 41](#_Toc293914932)

[All Files Visible 41](#_Toc293914933)

[Create 41](#_Toc293914934)

[Create Interface with AutoRun Pro 41](#_Toc293914935)

[Download and Install the Application 41](#_Toc293914936)

[Create a New Project 41](#_Toc293914937)

[Start the Application 41](#_Toc293914938)

[Make New Project 41](#_Toc293914939)

[Customize the template 44](#_Toc293914940)

[Apply logo 44](#_Toc293914941)

[Place PipeLine media files in a folder 45](#_Toc293914942)

[Create menu choices 45](#_Toc293914943)

[Create menu options that calls the PipeLine installer to run 46](#_Toc293914944)

[Create CD browse menu option 49](#_Toc293914945)

[Create link to external website menu option 50](#_Toc293914946)

[Create link to next page menu option 51](#_Toc293914947)

[Create second page 52](#_Toc293914948)

[Create open PipeLine User Guide PDF file in Adobe Acrobat 53](#_Toc293914949)

[Create Return to Main Menu 54](#_Toc293914950)

[Create Exit menu option 55](#_Toc293914951)

[Build the Executable AutoRun program and create the master CD 56](#_Toc293914952)

[Test the final master CD and make required number of CD copies 57](#_Toc293914953)

[Create Web Installer 58](#_Toc293914954)

[Web-download 65](#_Toc293914955)

[Application Design 67](#_Toc293914956)

[Overview 67](#_Toc293914957)

[Split Database Architecture 68](#_Toc293914958)

[Naming Conventions 68](#_Toc293914959)

[Data Types 69](#_Toc293914960)

[Graphical User Interface 70](#_Toc293914961)

[Explorer-Style Navigation 70](#_Toc293914962)

[Menu Bar Specification 71](#_Toc293914963)

[Form Conventions 72](#_Toc293914964)

[Control Use 74](#_Toc293914965)

[Command Buttons 74](#_Toc293914966)

[Report Conventions 75](#_Toc293914967)

[Report Layout 75](#_Toc293914968)

[Icons 76](#_Toc293914969)

[VBA Coding Guidelines 76](#_Toc293914970)

[Procedure/Function Declarations 76](#_Toc293914971)

[Variable Declarations 76](#_Toc293914972)

[Comments 76](#_Toc293914973)

[White Space 77](#_Toc293914974)

[Indentation 77](#_Toc293914975)

[Global Variables 77](#_Toc293914976)

[Database Schema 79](#_Toc293914977)

[Overview 79](#_Toc293914978)

[Entity Relationship Diagrams 80](#_Toc293914979)

[Table and Column Descriptions 81](#_Toc293914980)

[Program Units 120](#_Toc293914981)

[Overview 120](#_Toc293914982)

[Source Code CD content 120](#_Toc293914983)

[Program name 122](#_Toc293914984)

[Forms 123](#_Toc293914985)

[Reports 173](#_Toc293914986)

[Modules 190](#_Toc293914987)

[Menu Structure 217](#_Toc293914988)

[Overview 217](#_Toc293914989)

[Program calls by Menu choices 217](#_Toc293914990)

[Archiving and Backup 219](#_Toc293914991)

[Overview 219](#_Toc293914992)

[Interfaces with Other Systems 221](#_Toc293914993)

[Overview 221](#_Toc293914994)

[Generic systems (Export of all program data) 221](#_Toc293914995)

[Other Pipeline Databases (import shipment data) 223](#_Toc293914996)

[E-Catelog (Import product data) 223](#_Toc293914997)

[Quantimed (import forecast consumption data) 224](#_Toc293914998)

[Supply Chain Manager (import actual consumption data) 225](#_Toc293914999)

[APPENDIX A: UPGRADING DATA FILES 227](#_Toc293915000)

[Overview 227](#_Toc293915001)

[Version Numbering 227](#_Toc293915002)

[Upgrade Procedure 227](#_Toc293915003)

[SQL Statements Used 227](#_Toc293915004)

[APPENDIX B: PIPELINE SUMMARY 229](#_Toc293915005)

[Overview 229](#_Toc293915006)

[Access the PipeLine Summary Module 229](#_Toc293915007)

[Selecting Programs 229](#_Toc293915008)

[Adding Programs 230](#_Toc293915009)

[Creating Reports and Graphs 231](#_Toc293915010)

[Shipment Costs by Supplier Report 231](#_Toc293915011)

[Shipment Order Report 232](#_Toc293915012)

[Consumption Graph/Export 233](#_Toc293915013)

[Graph Type 234](#_Toc293915014)

[Exporting Summary Consumption Graph Data 235](#_Toc293915015)

[Couple-Years of Protection (CYP) Graph 236](#_Toc293915016)

[Creating the CYP Graph 236](#_Toc293915017)

[Displaying the Summary CYP Graph 237](#_Toc293915018)

[Exporting the CYP Graph 238](#_Toc293915019)

[CYP Graph 238](#_Toc293915020)

[FPPMES 238](#_Toc293915021)

[Previewing and Printing a Report or Graph 238](#_Toc293915022)

[APPENDIX C: THE REDDICK VBA (RVBA) NAMING CONVENTIONS, VERSION 6.01 239](#_Toc293915023)

[Changes to the Conventions 239](#_Toc293915024)

[An Introduction to Hungarian 239](#_Toc293915025)

[Tags 240](#_Toc293915026)

[Variable tags 240](#_Toc293915027)

[Constructing Properties Names 241](#_Toc293915028)

[Collection Tags 241](#_Toc293915029)

[Constants 242](#_Toc293915030)

[Menu Items 242](#_Toc293915031)

[Creating Data Types 242](#_Toc293915032)

[Enumerated types 242](#_Toc293915033)

[Tags for classes and user-defined types 243](#_Toc293915034)

[Polymorphism 243](#_Toc293915035)

[Constructing Procedures 244](#_Toc293915036)

[Constructing Procedure Names 244](#_Toc293915037)

[Naming Parameters 244](#_Toc293915038)

[Naming Labels 244](#_Toc293915039)

[Prefixes 244](#_Toc293915040)

[Arrays of Objects Prefix 244](#_Toc293915041)

[Index Prefix 245](#_Toc293915042)

[Prefixes for Scope and Lifetime 245](#_Toc293915043)

[Other Prefixes 245](#_Toc293915044)

[Suffixes 246](#_Toc293915045)

[File Names 246](#_Toc293915046)

[Host Application and Component Extensions to the Conventions 246](#_Toc293915047)

[Access 2000, Version 9.0 Objects 246](#_Toc293915048)

[DAO 3.6 Objects 248](#_Toc293915049)

[Visual Basic 6.0 Objects 251](#_Toc293915050)

[Microsoft ActiveX Data Objects 2.1 Tags 252](#_Toc293915051)

[Microsoft ADO Ext. 2.1 for DDL and Security (ADOX) Tags 252](#_Toc293915052)

[Microsoft Jet and Replication Objects 2.1 253](#_Toc293915053)

[Microsoft SQL Server and Microsoft Data Engine (MSDE) Objects 253](#_Toc293915054)

[Microsoft Common Control Objects 254](#_Toc293915055)

[Other Custom Controls and Objects 255](#_Toc293915056)

[Summary 256](#_Toc293915057)

[APPENDIX D: THE REDDICK VBA (RVBA) CODING CONVENTIONS (VERSION 0.90) 257](#_Toc293915058)

[Arrays 257](#_Toc293915059)

[Assertions 258](#_Toc293915060)

[Comments 259](#_Toc293915061)

[Constants 260](#_Toc293915062)

[Date Functions and Date Variables 261](#_Toc293915063)

[Default Properties 261](#_Toc293915064)

[Deprecated Features 261](#_Toc293915065)

[Disambiguation 262](#_Toc293915066)

[DLL Base Address [VB6] 262](#_Toc293915067)

[Dollar Sign ($) Functions 263](#_Toc293915068)

[Error Handling 263](#_Toc293915069)

[Exiting a Procedure 264](#_Toc293915070)

[For/Next and For Each/Next Loops 265](#_Toc293915071)

[GoTo Statements 265](#_Toc293915072)

[Headers 265](#_Toc293915073)

[Indenting 266](#_Toc293915074)

[Instantiation 267](#_Toc293915075)

[Labels 267](#_Toc293915076)

[Long Lines of Code 267](#_Toc293915077)

[Nothing 268](#_Toc293915078)

[Parameters to a Procedure 269](#_Toc293915079)

[Parentheses 270](#_Toc293915080)

[Procedure Scope 270](#_Toc293915081)

[Project Properties [VB6] 270](#_Toc293915082)

[Raising Errors 271](#_Toc293915083)

[Select/End Select Blocks 272](#_Toc293915084)

[Source Code Control [VB6] 272](#_Toc293915085)

[Type Conversion 273](#_Toc293915086)

[Variable Declaration 273](#_Toc293915087)

[Variable Initialization 274](#_Toc293915088)

[Variables Scope and Lifetime 274](#_Toc293915089)

[Version Compatibility [VB6] 274](#_Toc293915090)

[APPENDIX E: MICROSOFT APPLICATION USER INTERFACE GUIDELINES 277](#_Toc293915091)

[Intro 277](#_Toc293915092)

[General Properties 277](#_Toc293915093)

[Fonts 277](#_Toc293915094)

[Face 277](#_Toc293915095)

[Size 277](#_Toc293915096)

[Text 277](#_Toc293915097)

[Customer/Client 277](#_Toc293915098)

[Page Title 277](#_Toc293915099)

[Type 277](#_Toc293915100)

[Color 277](#_Toc293915101)

[Background Color 277](#_Toc293915102)

[Case 278](#_Toc293915103)

[Color 278](#_Toc293915104)

[Palette 278](#_Toc293915105)

[Text 278](#_Toc293915106)

[Dimensions 279](#_Toc293915107)

[Screen 279](#_Toc293915108)

[Height 279](#_Toc293915109)

[Width 279](#_Toc293915110)

[Regions 279](#_Toc293915111)

[Header & Primary Navigation 279](#_Toc293915112)

[Secondary Navigation 279](#_Toc293915113)

[Content 279](#_Toc293915114)

[Footer 279](#_Toc293915115)

[Controls 279](#_Toc293915116)

[Buttons 279](#_Toc293915117)

[Text Boxes 279](#_Toc293915118)

[List Boxes 280](#_Toc293915119)

[Flow 280](#_Toc293915120)

[Usability 280](#_Toc293915121)

[Keyboard vs. Mouse 280](#_Toc293915122)

[Section 508 280](#_Toc293915123)

[Performance 280](#_Toc293915124)

[Internationalization 280](#_Toc293915125)

[Multi-Language support 280](#_Toc293915126)

[Date & Time 280](#_Toc293915127)

[Applications 280](#_Toc293915128)

[Desktop 280](#_Toc293915129)

[Page 281](#_Toc293915130)

[Splash Screen 281](#_Toc293915131)

[Framework 281](#_Toc293915132)

[Title Bar 281](#_Toc293915133)

[Font 281](#_Toc293915134)

[Color 281](#_Toc293915135)

[Content 281](#_Toc293915136)

[Case 281](#_Toc293915137)

[Menu Bar 281](#_Toc293915138)

[Tool Bars 281](#_Toc293915139)

[Icon Size 281](#_Toc293915140)

[Spacing 281](#_Toc293915141)

[Status Bar 281](#_Toc293915142)

[Sizing 281](#_Toc293915143)

[Layout 281](#_Toc293915144)

[General 281](#_Toc293915145)

[Size 282](#_Toc293915146)

[Font 282](#_Toc293915147)

[Color 282](#_Toc293915148)

[Graphics 282](#_Toc293915149)

[Splash Screen 282](#_Toc293915150)

[About 282](#_Toc293915151)

[Regions 282](#_Toc293915152)

[Tree View 283](#_Toc293915153)

[Icons 283](#_Toc293915154)

[Form 283](#_Toc293915155)

[Report 283](#_Toc293915156)

[Question 283](#_Toc293915157)

[Configuration 283](#_Toc293915158)

[Group/Section 283](#_Toc293915159)

[Forms 283](#_Toc293915160)

[Layout 283](#_Toc293915161)

[Logos 283](#_Toc293915162)

[Style (Special Effect) 283](#_Toc293915163)

[Elements 283](#_Toc293915164)

[Title Bar 283](#_Toc293915165)

[Title 283](#_Toc293915166)

[Menu Bar 283](#_Toc293915167)

[Tool Bar 284](#_Toc293915168)

[Labels 284](#_Toc293915169)

[Pop Up Dialogs 284](#_Toc293915170)

[Modal 284](#_Toc293915171)

[When to use 284](#_Toc293915172)

[Buttons 284](#_Toc293915173)

[Modeless 284](#_Toc293915174)

[When to use 284](#_Toc293915175)

[Buttons 284](#_Toc293915176)

[Tabbed 284](#_Toc293915177)

[When to use 284](#_Toc293915178)

[Layout 284](#_Toc293915179)

[Buttons 284](#_Toc293915180)

[Elements 284](#_Toc293915181)

[Title Bar 284](#_Toc293915182)

[Title 284](#_Toc293915183)

[Menu Bar 284](#_Toc293915184)

[Tool Bar 284](#_Toc293915185)

[Labels 285](#_Toc293915186)

[Style (Special Effect) 285](#_Toc293915187)

[Reports 285](#_Toc293915188)

[Size 285](#_Toc293915189)

[DP-Body textFormat 285](#_Toc293915190)

[Paper Size 285](#_Toc293915191)

[DP-Body textMargins 285](#_Toc293915192)

[DP-Body textLayout 285](#_Toc293915193)

[Controls 285](#_Toc293915194)

[DP-Body textGeneral 285](#_Toc293915195)

[Dialog Base Units 285](#_Toc293915196)

[DP-Body textSize 285](#_Toc293915197)

[DP-Body textExamples: 285](#_Toc293915198)

[DP-Body textSpacing 285](#_Toc293915199)

[DP-Body textControls 285](#_Toc293915200)

[Dialogs 285](#_Toc293915201)

[Paragraphs 285](#_Toc293915202)

[Text Labels 285](#_Toc293915203)

[Group Boxes 285](#_Toc293915204)

[Buttons 285](#_Toc293915205)

[Grouping 285](#_Toc293915206)

[Color 286](#_Toc293915207)

[Other considerations 286](#_Toc293915208)

[Alignment 286](#_Toc293915209)

[Text box 286](#_Toc293915210)

[Layout 286](#_Toc293915211)

[Unlocked 286](#_Toc293915212)

[Font 286](#_Toc293915213)

[Color 286](#_Toc293915214)

[Text 286](#_Toc293915215)

[Background 286](#_Toc293915216)

[Size 286](#_Toc293915217)

[Locked 286](#_Toc293915218)

[Font 286](#_Toc293915219)

[Color 286](#_Toc293915220)

[Text 286](#_Toc293915221)

[Background 286](#_Toc293915222)

[Size 286](#_Toc293915223)

[List box 287](#_Toc293915224)

[Font 287](#_Toc293915225)

[Color 287](#_Toc293915226)

[Size 287](#_Toc293915227)

[Combo box 287](#_Toc293915228)

[Font 287](#_Toc293915229)

[Color 287](#_Toc293915230)

[Size 287](#_Toc293915231)

[Radio buttons 287](#_Toc293915232)

[When to use 287](#_Toc293915233)

[Colors 287](#_Toc293915234)

[Size 287](#_Toc293915235)

[Arrangement 287](#_Toc293915236)

[Check boxes 287](#_Toc293915237)

[When to use 287](#_Toc293915238)

[Colors 287](#_Toc293915239)

[Size 287](#_Toc293915240)

[Arrangement 287](#_Toc293915241)

[Command Buttons 287](#_Toc293915242)

[Size 287](#_Toc293915243)

[Behavior 288](#_Toc293915244)

[Ok 288](#_Toc293915245)

[Cancel 288](#_Toc293915246)

[Apply 288](#_Toc293915247)

[Clear 288](#_Toc293915248)

[Next 288](#_Toc293915249)

[Previous 288](#_Toc293915250)

[Help 288](#_Toc293915251)

Figures

[Figure 1 - PipeLine Process Flow 25](#_Toc293915252)

[Figure 2 - PipeLine References 33](#_Toc293915253)

[Figure 3 - Application Information 34](#_Toc293915254)

[Figure 4 - Application Files 34](#_Toc293915255)

[Figure 5 - Extra Information 35](#_Toc293915256)

[Figure 6 - Runtime Files 36](#_Toc293915257)

[Figure 7 - Support Files 37](#_Toc293915258)

[Figure 8 - Additional Registry Keys 38](#_Toc293915259)

[Figure 9 - Welcome Dialog 39](#_Toc293915260)

[Figure 10 - Merge Modules 40](#_Toc293915261)

[Figure 11 - Build Options 40](#_Toc293915262)

[Figure 12 - File Dropdown Menu 42](#_Toc293915263)

[Figure 13 - Project Options 42](#_Toc293915264)

[Figure 14 – Main Page 43](#_Toc293915265)

[Figure 15 – Page Templates 43](#_Toc293915266)

[Figure 16 - Properties Window 44](#_Toc293915267)

[Figure 17 - Image Editor 45](#_Toc293915268)

[Figure 18 - PipeLine Files 45](#_Toc293915269)

[Figure 19 - Visual Layout 46](#_Toc293915270)

[Figure 20 - Main Menu Properties 47](#_Toc293915271)

[Figure 21 - OnClick Actions 48](#_Toc293915272)

[Figure 22 - File Selection Window 48](#_Toc293915273)

[Figure 23 – Available OnClick Events 49](#_Toc293915274)

[Figure 24 – Browse CD OnClick Action 50](#_Toc293915275)

[Figure 25 - USAID|DELIVER PROJECT Website OnClick Action 51](#_Toc293915276)

[Figure 26 - Documentation OnClick Action 52](#_Toc293915277)

[Figure 27 - Duplicate Page 53](#_Toc293915278)

[Figure 28 - User Guide OnClick Action 54](#_Toc293915279)

[Figure 29 - Return to Main Menu OnClick Action 55](#_Toc293915280)

[Figure 30 - Exit OnClick Action 56](#_Toc293915281)

[Figure 31 - File Dropdown Menu 56](#_Toc293915282)

[Figure 32 - Publish Project 57](#_Toc293915283)

[Figure 33 - Executable File Information 57](#_Toc293915284)

[Figure 34 - Final Auto-run CD 58](#_Toc293915285)

[Figure 35 - WinZip Self Extractor Welcome Window 59](#_Toc293915286)

[Figure 36 - WinZip Self Extractor Select Type Window 59](#_Toc293915287)

[Figure 37 - WinZip Self Extractor Span Options Window 60](#_Toc293915288)

[Figure 38 - WinZip Self Extractor File Selection Window 60](#_Toc293915289)

[Figure 39 - WinZip Self Extractor Password Notification 61](#_Toc293915290)

[Figure 40 - WinZip Self Extractor Message Text Option Window 61](#_Toc293915291)

[Figure 41 - WinZip Self Extractor "Unzip To" Option Window 62](#_Toc293915292)

[Figure 42 - WinZip Self Extractor Command Options Window 62](#_Toc293915293)

[Figure 43 - WinZip Self Extractor About Box Options Window 63](#_Toc293915294)

[Figure 44 - WinZip Self Extractor Icon Selection Window 63](#_Toc293915295)

[Figure 45 - WinZip Self Extractor Miscellaneous Options Window 64](#_Toc293915296)

[Figure 46 - WinZip Self Extractor Ready to Create Window 64](#_Toc293915297)

[Figure 47 - WinZip Self Extractor Test Window 65](#_Toc293915298)

[Figure 48 – Sample Menu 71](#_Toc293915299)

[Figure 49 – Sample Detail Form 72](#_Toc293915300)

[Figure 50 - ERD Diagram 80](#_Toc293915301)

[Figure 51 - xml Export file schema diagram 221](#_Toc293915302)

[Figure 52 - xml Export file schema diagram 222](#_Toc293915303)

[Figure 53 - xml Export file schema diagram 223](#_Toc293915304)

[Figure 54 - xml Export file schema diagram 224](#_Toc293915305)

[Figure 55 - xml Export file schema diagram 225](#_Toc293915306)

[Figure 56: xml Export file schema diagram 226](#_Toc293915307)

[Figure 57 - Select Program 230](#_Toc293915308)

[Figure 58 - Enter Program ID window 230](#_Toc293915309)

[Figure 59 - — PipeLine Summary Shipment Costs by Supplier screen 232](#_Toc293915310)

[Figure 60 - PipeLine Summary Shipment Order Report screen 233](#_Toc293915311)

[Figure 61 - PipeLine Summary Consumption Graph screen 234](#_Toc293915312)

[Figure 62 - Graph options 234](#_Toc293915313)

[Figure 63 - Forecast options 235](#_Toc293915314)

[Figure 64 - Output window 235](#_Toc293915315)

[Figure 65 - PipeLine Summary CYP graph screen 236](#_Toc293915316)

[Figure 66 - Graph type selection 237](#_Toc293915317)

[Figure 67 - CYP Graph display 237](#_Toc293915318)

[Figure 69 - Export CYP Data Window 238](#_Toc293915319)

Tables

[Table 1 - List of Appendices 25](#_Toc293915320)

[Table 2 - Hardware and Software Requirements 26](#_Toc293915321)

[Table 3 - Installation Locations and Purpose 30](#_Toc293915322)

[Table 4 - Recommended Software Tools 32](#_Toc293915323)

[Table 5 – Prefixes for Access Database Window Objects 68](#_Toc293915324)

[Table 6 – Access Object Variable Prefixes 69](#_Toc293915325)

[Table 7 – Field Name Conventions 69](#_Toc293915326)

[Table 8 – Data Types 70](#_Toc293915327)

[Table 9 – Menu Element Specification Table 71](#_Toc293915328)

[Table 10 – Standard functionality of Switchboard Strip Menus 71](#_Toc293915329)

[Table 11 - Element Styles 73](#_Toc293915330)

[Table 12 – Control rules 74](#_Toc293915331)

[Table 13 – Functionality of standard buttons 74](#_Toc293915332)

[Table 14 – Report Checklist 75](#_Toc293915333)

[Table 15 - List of Tables 81](#_Toc293915334)

[Table 16 - Source Code CD Content 120](#_Toc293915335)

[Table 17 - List of Forms 123](#_Toc293915336)

[Table 18 - List of Reports 173](#_Toc293915337)

[Table 19 - List of Modules 190](#_Toc293915338)

[Table 20 - Program Calls by Menu Choices 217](#_Toc293915339)

[Table 21 - Tables for VBA Variables 241](#_Toc293915340)

[Table 22 - Scope prefixes 245](#_Toc293915341)

[Table 23 - Other commonly-used prefixes 245](#_Toc293915342)

[Table 24 - Commonly-used suffixes 246](#_Toc293915343)

[Table 25 - Access object variable tags 247](#_Toc293915344)

[Table 26 - DAO object tags 248](#_Toc293915345)

[Table 27 - Access Database Explorer object tags 249](#_Toc293915346)

[Table 28 - Specific object tags and suffixes for Access Database Explorer objects 249](#_Toc293915347)

[Table 29 - Field tags (if you decide to use them) 250](#_Toc293915348)

[Table 30 - Visual Basic 6.0 object tags 251](#_Toc293915349)

[Table 31 - ADO 2.1 Object tags 252](#_Toc293915350)

[Table 32 - ADOX Object tags 253](#_Toc293915351)

[Table 33 - JRO object tags 253](#_Toc293915352)

[Table 34 - SQL Server/MSDE object tags 254](#_Toc293915353)

[Table 35 - Microsoft Common Control Object tags 254](#_Toc293915354)

[Table 36 - Tags for commonly-used custom controls 255](#_Toc293915355)

Acronyms

ADO ActiveX Data Object

AIDS Acquired Immune Deficiency Syndrome

DLL Dynamic Link Library

ERD Entity Relationship Diagram

FK Foreign Key

NGO Non-Governmental Organization

PK Primary Key

SCMgr Supply Chain Manager

SDG Software Development Group

USAID U.S. Agency for International Development

VB Visual Basic

VBA Visual Basic for Applications

Acknowledgments

The author would like to thank Jeffrey Leiner, Muhammad Ahmed, Ashraf Islam, Trisha Long, and Joy Kamunyori for their helpful suggestions and expert reviews of this document and the developers who worked on PipeLine and took the time to actually document the code making this effort easier (Muhammad Ahmed, Lisa Blankenship, Jeff Leiner, and Mike Tracy).

The author would also like to Acknowledge USAID for funding this effort and those from FPLM and DELIVER projects over the years who championed PipeLine development such as Trisha Long, Karen Ampeh, John Durgavich and the countless others who contributed in so many ways.

Introduction

Overview

The Pipeline Monitoring and Procurement Planning System (PipeLine), a software tool, was designed to help program managers monitor the status of their product pipelines and product procurement plans. PipeLine provides information needed to initiate and follow-up actions to ensure the regular and consistent stock of products at the program or national level. Consistency of stock is the first step in meeting the basic objective of any logistics system, which is to provide—

* the *right* quantities
* of the *right* commodities
* in the *right* condition
* in the *right* place
* at the *right* time

for the *right* cost.

These are the *six rights* of logistics management.

What PipeLine Can Do for You

PipeLine helps you achieve the *right* quantities at the *right* time.

For each product, PipeLine monitors—

* Total quantities *consumed* (i.e., amounts dispensed to users or sold to clients)
* *Shipments* of new products (planned, ordered, shipped, or received) into your program
* and the values of your products
* *Inventory levels* for each product in your program's logistics system (desired and actual)
* *Inventory level changes* (e.g., product losses or transfers out of or into your program)

With these data and an understanding of the *lead time* required for each step in the procurement process, PipeLine can—

1. Show what actions you need to take for procurement planning and management, and when these actions should be taken.
2. Identify impending problems (i.e., surpluses, shortfalls, or stockouts) *before* they occur.
3. Calculate procurement quantities needed to keep your pipeline in balance.
4. Calculate the estimated value of shipments or maintain the actual value (if known).

You can use this information with program policymakers, product suppliers, and donors to provide a rational basis for planning future product needs.

PipeLine is *not* the answer to every logistics question. It helps monitor the *aggregate* quantity of each product entering and leaving your program's distribution system (preferably using data from a logistics management information system [LMIS]).

PipeLine's utility is enhanced if your program has a well-functioning LMIS and forecasting procedures. Even without these underlying systems, use PipeLine with whatever data are available. By beginning a rational and systematic product monitoring and planning process, you take the first step toward ensuring consistent stock levels.

Why Use PipeLine?

Ensuring adequate supplies of commodities is difficult for most programs. As a program manager, you face a complex procurement planning environment, characterized by—

1. Multiple suppliers of many products (local and private suppliers, bilateral and multilateral donors, etc.), each with its own products, lead times, costs, information needs, and bureaucratic constraints
2. Proliferation of service delivery points, in many cases in an integrated service delivery setting, and/or with multiple service delivery organizations served by a single logistics system
3. Increasing volume (and costs) of commodities, which must be managed and moved through complex distribution channels
4. Increasing emphasis on accountability, cost-effectiveness, and sustainability from donors who fund product procurement and from policymakers.

You need to monitor the quantity and timing of multiple products entering your logistics system from multiple suppliers. Because procurement lead times may be long—years, in many cases— you need to take action months or years before commodities are needed to receive them on time.

You may need to negotiate with many different suppliers and donors to obtain the quantities you require. Such negotiations are best accomplished when specific data on product requirements are available. You must know when you will stock out of each product, how much must be procured to meet future needs, and when you should receive it. To prevent overordering, you must also know what quantities would exceed your storage capacity or risk wastage due to expiry. PipeLine can provide this information.

Who Should Use PipeLine?

In a multi-product, multi-supplier environment, procurement planning and pipeline monitoring functions cannot be donor driven. It is increasingly necessary that local program managers be empowered to do their own forecasting, pipeline monitoring, and procurement planning; they must also take charge of coordinating the activities of donors and local suppliers, as well as those of their own logistics management staff. Donor staff often have other priorities and little time to devote to the details of logistics management. Commercial suppliers have interests that may or may not correspond to the interests of your organization.

If you are the logistics manager or program manager for your organization, you should manage your own pipeline. PipeLine can help.

While your managers and decision makers will be the primary users of PipeLine, the system can provide information to—

**Suppliers of commodities**PipeLine provides reports on the current status and the cost of pending shipments from aspecific supplier, which that supplier can use to monitor product flow.

**Purchasers/donors of commodities**Staff who finance the purchase of commodities can use PipeLine reports and graphs to understand the current pipeline status and future requirements.

**Host-country policymakers**PipeLine reports and graphs can be used to help policymakers understand issues with the levels of particular commodities and the implications of different decisions on the availability of the product.

PipeLine Software Functions

PipeLine can help you with pipeline monitoring and procurement planning functions, as explained below.

Pipeline Monitoring

Pipeline monitoring functions include—

* Monitoring stock balances, in terms of quantities and months of stock on hand in the entire program (aggregate of stock at all levels)
* Comparing stock balances to maximum and minimum stock policies
* Automating the identification of pipeline problems (quantities needed, stockouts, balances below minimum or above maximum)

Providing couple-years of protection (CYP) conversion graphs.

Procurement Planning

Procurement planning functions include—

* Calculation of shortfalls/surpluses and quantities needed to maintain the program’s desired stock levels
* Automated calculation and tracking of pending pipeline actions, based on lead times (shipments to plan, order, ship, and receive)
* Application of USAID’s contraceptive procurement tables (CPT) format for the computation of calendar year quantities required and the generation of data for USAID’s planning requirements
* Calculation of estimated costs of shipments and freight
* Comparison of alternative procurement scenarios and analysis
* Alternative unit of measure calculation displays products in Basic Units. Basic Units are used to quantify patient or consumer needs and usually refers to tablets, capsules, or milliliters, rather than packs or bottles.

Technical Architecture

The PipeLine software is developed using following programming language, tools and techniques:

* Microsoft Access 2003 and 2000. The front end database is based on Access 2003 and the back-end database is based on Access 2000.
* On-line help was developed using Robohelp X5
* Automated installable version was created using InstallShield 12 and SageKey for Access 2003 version 2.0.9
* Data export/import interface between Supply Chain Manager and PipeLine is build using XML 1.1
* CD auto-run was build using AutoRun Pro Enterprise II
* User Guide and Technical documentation was developed using combination of Microsoft Visio, Powerpoint and Word
* For source control, Microsoft Visual Source Safe was used.

For bug tracking and issue tracking, open source bugzilla application was used.

Process Flows

Figure - PipeLine Process Flow

**Update**

**Background  
Data**

**Run Imports**

(i.e. Product  
Catalog from  
ORION, Shipment,  
and Consumption  
Data)

**Update  
Shipment Data**

**Update/Import  
Consumption Data**

**Run Forecast**

**Balance PipeLine**(i.e. adjust shipment schedules as needed)

**Run Reports**

**Update Stock Data**

**PIPELINE PROCESS**

Appendices

Table - List of Appendices

|  |  |  |
| --- | --- | --- |
| Appendix | Title | Description |
| A | The Reddick VBA (RVBA) Naming Conventions, Version 6.01 | Industry standard naming conventions for Access applications |
| B | The Reddick VBA (RVBA) Coding Conventions (version 0.90) | Industry standard coding conventions for Access applications |
| C | Microsoft Application User Interface Guidelines | JSI GUI standards |

System Requirements

The following resources are recommended for use with PipeLine—

Table - Hardware and Software Requirements

|  |  |
| --- | --- |
| Component | Hardware/Software Requirements |
| CPU | Pentium IV or higher |
| Operating System | Windows XP or above |
| Memory | 1 GB or higher |
| Hard Disk Space | 500 MB of free space or higher |
| Video Adapter | SVGA with at least 800 X 600 resolution |
| Microsoft Office | Microsoft Office 2003 or higher |

Installation and Configuration

Installation Instructions

Microsoft Office® 2003

Although PipeLine will run without Microsoft Office® 2003 installed on your computer, having Office 2003 installed will enhance PipeLine’s usefulness by allowing PipeLine to export data files to Word® or Excel®.

How to Install PipeLine

PipeLine can be installed from a CD-ROM or the Internet.

Before You Begin

You can run PipeLine 2 and PipeLine 5. on the same computer, but we recommend that you uninstall PipeLine 2 BEFORE installing PipeLine 5.

To uninstall PipeLine 2—

1. Click on Start.

Previous Versions

All other previous versions of PipeLine cannot be run on the same computer as PipeLine 5. When you install PipeLine 5, the installer will automatically remove the previous version of PipeLine prior to installing PipeLine 5.

1. Click on the Settings option.
2. Click on the Control Panel option.

After the Control Panel window opens—

1. Click on the Add/Remove Programs option.

Locate and click on PipeLine in the Currently Installed Programs list.

1. Click on the Change/Remove button.

The PipeLine 2 setup program will start, and will prepare your computer to uninstall PipeLine 2.

1. Click on the Remove All Button.

A message is displayed asking if you want to remove PipeLine.

Data Files

Data files created with previous versions of PipeLine are not removed from your system. See Converting Your Existing Data Files on page 17 for information on converting your existing PipeLine 2 data files.

1. Click on the Yes button to begin the uninstall procedure.

When the uninstall procedure is completed, you will be prompted to restart windows.

1. Click on the Restart Windows button.

Installing PipeLine from a CD

1. Start Microsoft Windows.
2. Insert the PipeLine CD.

The PipeLine installation should begin automatically.

1. Follow the on-screen instructions.
2. If the installation does not begin automatically—
   1. Click on Start on the Windows Taskbar.
   2. Click on Run from the pop-up menu.
   3. In the Command Line box, type *x:setup* ("x" is the letter of your CD-ROM drive).
   4. Click on the OK button, and follow the on-screen instructions.

After PipeLine is successfully installed, the PipeLine shortcut (shown below) will be displayed on your desktop.

Installing PipeLine from the Internet

PipeLine CD-Rom

If your Internet connection is slow and/or unreliable, order the PipeLine CD-ROM.

PipeLine is available on the USAID | DELIVER PROJECT website at the following web address:

http://deliver.jsi.com.

To download PipeLine—

1. Access the Internet, and enter the USAID | DELIVER PROJECT web address.
2. Locate the PipeLine download page, and follow the on-screen instructions to download PipeLine.

How to Start PipeLine

PipeLine can be started from the Windows desktop or the Windows taskbar.

Starting PipeLine from the Windows desktop

From the Windows desktop—



1. Locate and double-click on the PipeLine icon to start the application.

Starting PipeLine from the Windows taskbar

From the Windows taskbar—

1. Click on Start.
2. Click on Programs.
3. Locate and click on the PipeLine 5.0 link.

Reinstalling PipeLine

To reinstall PipeLine—

Password

The Internet version of PipeLine requires a password to start the install process. That password was sent to you by email when you downloaded PipeLine. If you no longer have the email containing the PipeLine password, download PipeLine from the USAID | DELIVER PROJECT website: (http://deliver.jsi.com).

1. Place the PipeLine CD in your CD-ROM drive.

If you do not have the PipeLine CD, you can use the copy of PipeLine you downloaded from the USAID | DELIVER PROJECT website.

1. Start the install process, and follow the instructions on your screen.

During the process, a message box is displayed instructing you to remove PipeLine from your computer.

1. Click on the Remove button to remove PipeLine from your computer.
2. Click on the Finish button when prompted.

After PipeLine has been removed—

1. Repeat the PipeLine installation procedure.

See page for information on installing PipeLine.

Converting Your Existing Data Files

This version of PipeLine allows you to convert data created with previous versions of PipeLine.

Upgrading

The current program’s data file is not the current version. You can allow PipeLine to upgrade the file now. If you do not, some of PipeLine’s features may not work properly.

From the Program Data screen—

1. Click on the File Menu Bar option, and select the Open option from the pull-down menu.

PipeLine opens a window so you can locate the data you need to convert.

1. Locate and select the data file you need, and click on the Open button.

After you select the data file you need to convert, PipeLine displays a message similar to the one in the text box below.

1. Click on the Yes button to convert the selected data.

PipeLine opens a window, and allows you to rename the file you selected to upgrade. This safeguards the original data by saving the upgraded data under a different name.

Original Data

Remember, your original data remains in its original directory with its original name. The converted data is a copy of the original.

1. Type the new name in the File Name field, and click on the Open button.

PipeLine converts the selected data, renames the file, and displays its associated program data on the Program Data screen. You can now work with the converted data file.

Run-time installation

Table - Installation Locations and Purpose

|  |  |
| --- | --- |
| Directory/File | Purpose |
| **PipeLine** | **Parent directory** |
| **/ANYMOH** | **Directory for sample database** |
| /globalmoh.MDB | Sample Database |
| **/Data** | **Directory for live databases** |
| **/Graphics** | **Directory for application graphics** |
| /SplashNewT.avi | Splash screen movie |
| /PL40.ico | PipeLine taskbar icon |
| /Pipeline\_ICON-xx.ico | PipeLine Desktop icon |
| **/Import** | **Directory for imported files** |
| **/Summary** | **Directory for PipeLine Summary** |
| /Roboex32.dll | Dll required for PipeLine Summary to run properly |
| /Proc2000.mdb | PipeLine Summary frontend |
| /Proc\_BE.mdb | PipeLine Summary backend |
| /Prog2000.mdb | PipeLine Summary program list |
| /Summary.ico | PipeLine Summary icon |
| /Sumv2.cnt | PipeLine Summary help cnt file |
| /SUMv2.hlp | PipeLine Summary help file |
| **/XML** | **Directory for xml files** |
| /ECatalog\_Live\_Final\_Generic\_20100701.xml | E-Catalog file distributed with application |
| /SCMS Product\_ARV\_TEST.xml | SCMS ARV file distributed with application |
| /Contraceptives.xml | Contraceptives file distributed with application |
| /e-help.cnt | PipeLine (English) help cnt file |
| /e-help.HLP | PipeLine (English) help file |
| /E-PL-help.cnt | PipeLine help cnt file |
| /E-PL-help.hlp | PipeLine help file |
| /Pipeline\_ICON-xx.ico | PipeLine icon |
| /Pipeline2000.MDB | PipeLine frontend file |
| /PLFix1.reg | Registry fix for graphs |
| /PLFix2.reg | Registry fix for graphs |
| /PLFix3.reg | Registry fix for graphs |
| /pmp\_mpty.mdb | Empty PipeLine backend file |
| /posttransform.xslt |  |
| /ProgV4.mdb | PipeLine program list |
| /ReadMe.txt | Readme file for installation issues and known issues |
| /Roboex32.dll | Dll required for PipeLine to run properly |

Development environment installation

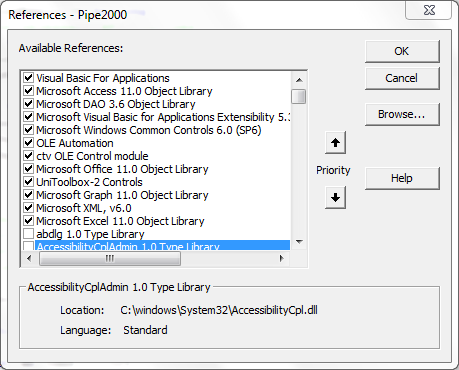
The Development Environment requires the installation of Microsoft Office 2003 (MSACCESS and Excel are required). Some other tools are suggested as well

Table - Recommended Software Tools

|  |  |  |
| --- | --- | --- |
| Software/Tool | Required | Description |
| Microsoft Access | Y | The Main Development Tool for PipeLine |
| Microsoft Excel | Y | Required for the creation of the Output to Excel reports |
| Microsoft Visual SourceSafe (VSS) | N | The software code repository for PipeLine. Required for checking in/out the source code files. Optional (but suggested) if development is being done in a stand alone environment |
| Microsoft Access Plug-In: Source Code Control | N | odc\_accscc.exe, this allows VSS integration into Access. Access .mdb files can be stored in VSS as individual components allowing multi-user development on a single .mdb file). |
| FMS Total Visual Code Tools | N | This plugin/toolbar provides the developer with the ability to quickly stub in new procedures and functions and to cleanup existing code modules with proper development standards. |
| Microsoft Windows Common Controls 6.0 (SP6) | Yes | This provides the Treeview control. |

The Access References should look like this:

Figure - PipeLine References



Build process

The Build Process for PipeLine involves checking the code into VSS (if necessary), removing the application for VSS, compact the application, and updating the tblSysParameters. In the tblSysParameters, update the AsOf date, the Version number, and set InitialInstall to True. Please note that the application will not relink properly if InitialInstall is not set to true since this flag informs the application to look for the default data file distributed with PipeLine.

Installer

PipeLine is widely used in many countries. It is also downloaded through the web. A majority of the users install the PipeLine application on their own. The installer is first created using SageKey MSI Wizard 2003 along with Installshield 12. In order to then provide a simple, user friendly and self-installable interface, a tool called AutoRun Pro Enterprise II was used. (This tool is available at this website <http://www.longtion.com>.) These file create the cd image and the files are zipped using WinZip Self Extractor to create a single downloadable file used for the web distribution.

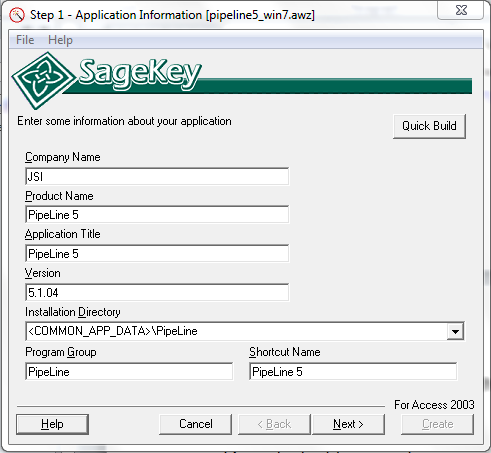
Mechanics for Creating the CD Installer

Create files with SageKey

Enter Application Information

Once the build is complete, open the file PipeLine5\_win7.awz.

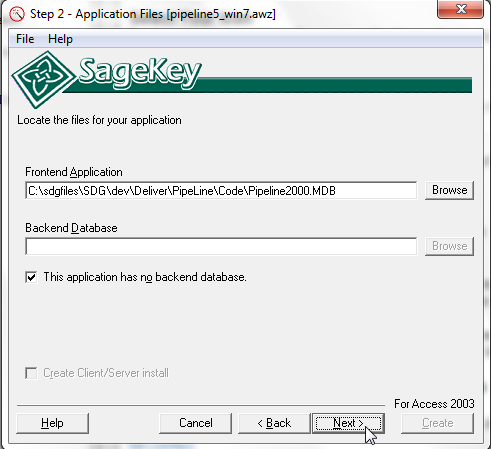
Figure - Application Information



Select Application Files

Update the basic information about PipeLine here. For more information on each field, right click the corresponding label and select 'What's this?" Click Next to go to next screen.

Figure - Application Files



Select your database project files here and click Next to go to next screen.

Front-end Application

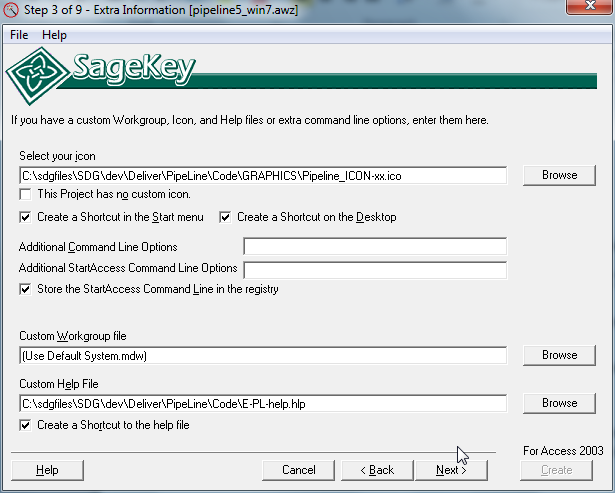
Click 'Browse' to locate the PipeLine2000.mdb file.  The files must exist in the place specified or you can not proceed to the next step.

Back-end Database

Since we connect to the backend dynamically, check the box next to 'This application has no back-end database.' The text field will go blank and the browse button will no longer be available.

Enter Extra Information

Figure - Extra Information



Select Your Icon

Browse to the PipeLine icon found in the graphics folder. The file must exist to proceed to the next step.

Select Shortcuts

Choose the Desktop and the Start Menu shortcuts.

Store the StartAccess Command Line in the registry & System.mdw

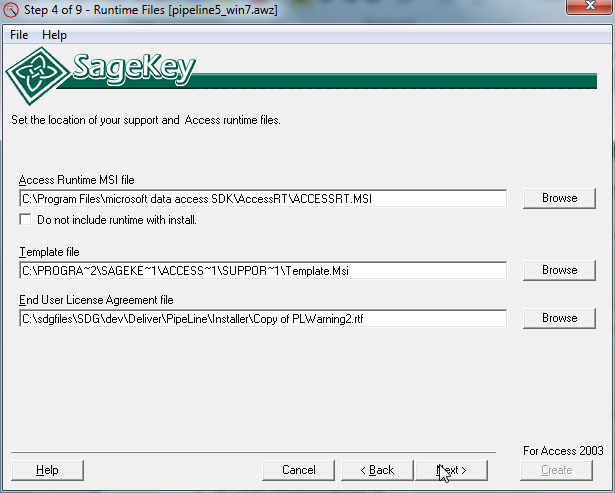
Choose to stre the startaccess command line in the registry. And use the default system.mdw

Custom Help File

Browse to the PipeLine help file. If you want a shortcut to the help file on the Start Menu check the box.

Select Runtime Files

Figure - Runtime Files



Access Runtime Files:

Browse to location of the 'Accessrt.msi'.

Template File:

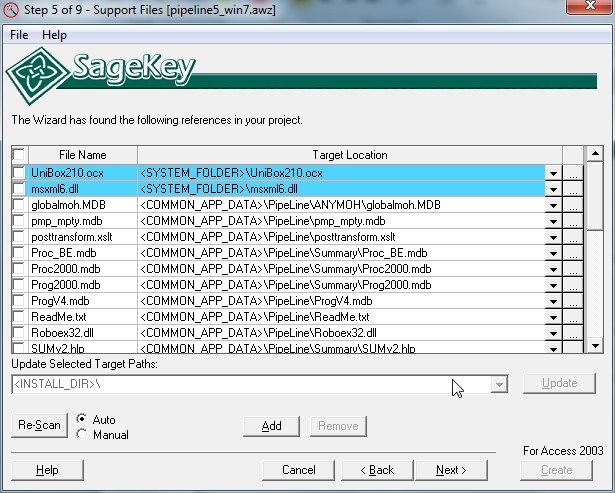
Browse to the directory that contains the wizard support files and locate the file "Template.msi". (The installation of the Wizard will set this field to the default directory \Program Files\SageKey Software\Access 2003 MSI Wizard\Support Files\Template.msi)

End User License Agreement file:

Browse ot the PipeLine Warning file in the installer directory.

Select Other Files

Figure - Support Files



When you click 'Next' in Step 4 (Runtime Files), the Wizard opens the file you specified as your Front-end in Step 2 (Application Files) and scans it for references. If it finds any, they are added here. You can remove any erroneous ones, or, if you make changes at a later time, click the 'Re-Scan' button to refresh the list.

The Wizard will try to determine the target path for each of these, but you can change the installation path by editing the Target Path combo field.  You can edit this and change it, but do not change or add new variables. (i.e.: Don't change or remove <PROGRAM\_FILES> or any other words enclosed in less than (<) or greater than (>) brackets.) This will cause the Installation build to fail.

Any additional files your application needs that are not included as references are added here as well. Clicking the 'Add' button will bring up the Add File dialog.

You can select multiple files to add in this dialog.  Once you have selected the file or files you want to add to the install, click the Open button to add these files.

By Default, each file you selected will have a target directory of <INSTALL\_DIR>\filename.  You can change the target directory for multiple files by checking the checkbox beside each file to change and entering a new target path in the "Update Selected Target Path" combo box.  Press the Update button to update the target path of the selected files.

File Details

At the right hand side of each file row, there is a button which will display the File Details dialog.

This dialog allows you to specify the Source Path of the file in the case that you have moved the source files for you project.  Also, you can check the "Show Advanced Options" checkbox to view the advanced options for the specific file.

Please Note:  Changing any of the advanced fields is only recommended for users who are very familiar with msi.

**Attributes:**This is the Attributes column for the Component this file will belong to.  For more information, see:

<http://msdn2.microsoft.com/en-us/library/aa368007.aspx>

**Install Condition:** This is the Condition column for the Component this file will belong to.  For more information, see:

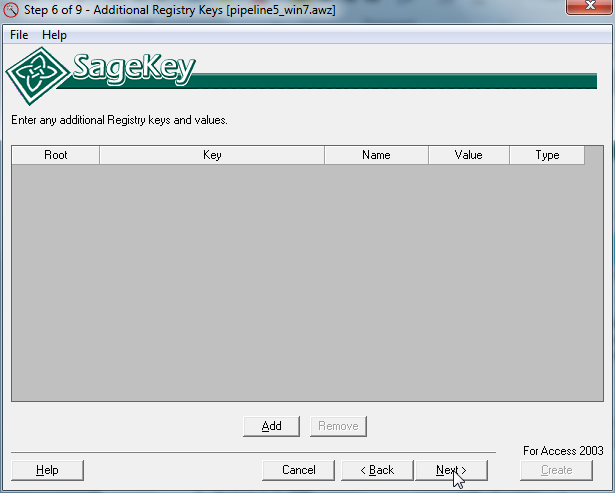
<http://msdn2.microsoft.com/en-us/library/aa368007.aspx>

Re-Scan

By Default, each time you go through the wizard for an existing project, the front-end database will be scanned for references.  This can be changed by selecting the "Manual" radio button beside the Re-Scan button.  If Manual is chosen, then the front-end database will not be automatically scanned for references.

Enter Registry Keys

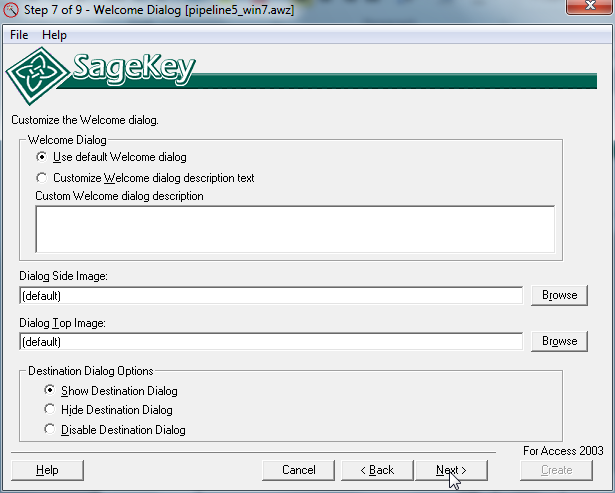
Figure - Additional Registry Keys



Since PipeLine does not need any special registry keys, click next to continue.

Design Welcome Dialog

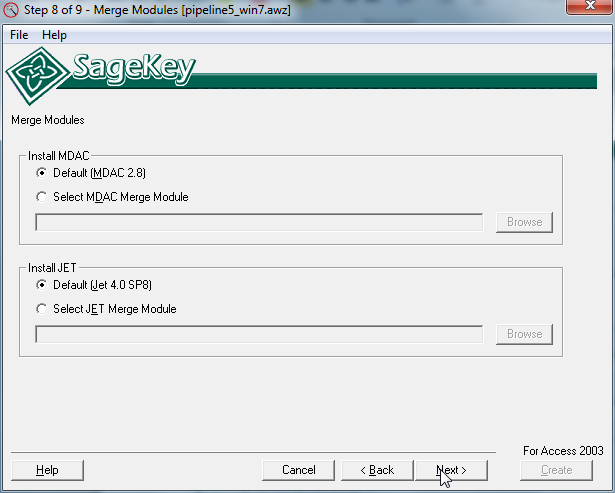
Figure - Welcome Dialog



Choose to use the default welcome dialog and show destination dialog and click Next.

Select Merge Modules

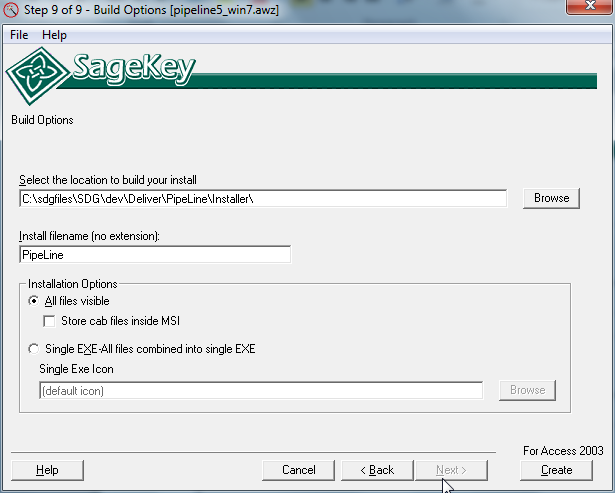
Figure - Merge Modules



Choose to use default MDAC and Jet and click next.

Enter Build Options

Figure - Build Options



Build Location:

Choose where the Wizard will build the Microsoft Windows Installer (.msi) file.

Install File Name

Type the name you wish to use for the main install file Do not type an extension for this file, it will be added by the Wizard.

Installation Options

All Files Visible

Select all files visble to have all of the files visible in the installation directory. You will need to include:

* -[InstallFilename].exe
* -Install.ini
* -the files subfolder

Create

Click the Create Button and the Wizard will verify if all the properties look correct, if so, it then builds the Install.

Create Interface with AutoRun Pro

Download and Install the Application

Download the application from the link below:

<http://www.longtion.com/autorunenterpriseii/autorunpro.htm>

Install the application. The installation steps are simple, like any other windows application, wizard driven, and takes under 2 minutes.

Once the application is installed, start the application and start building a project.

Create a New Project

Start the Application

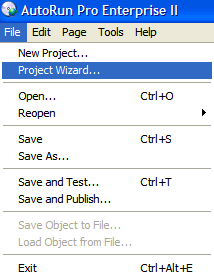
Find and click on the AutoRun Pro icon on your desktop



Make New Project

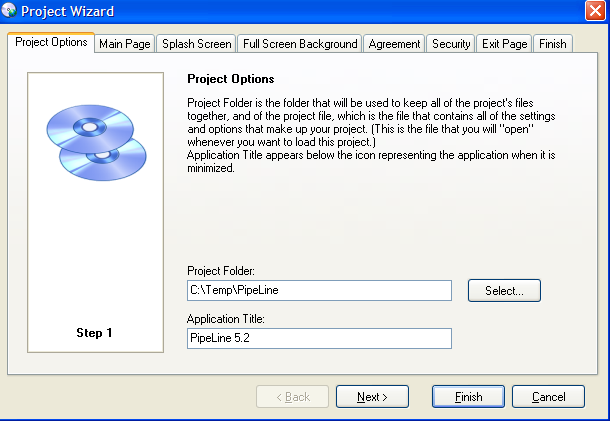
Create a new project to build the PipeLine CD auto-run program. See following illustrations for the steps.

Figure - File Dropdown Menu



Select File > Project Wizard. The Project Wizard starts.

Figure - Project Options



Select the folder on which to save the project files. Give a name to the project.

CD installer can display installation screens through multiple menu/pages. In next screen of the Project Wizard select a suitable template to base the look and feel of the installation screen.

Figure – Main Page

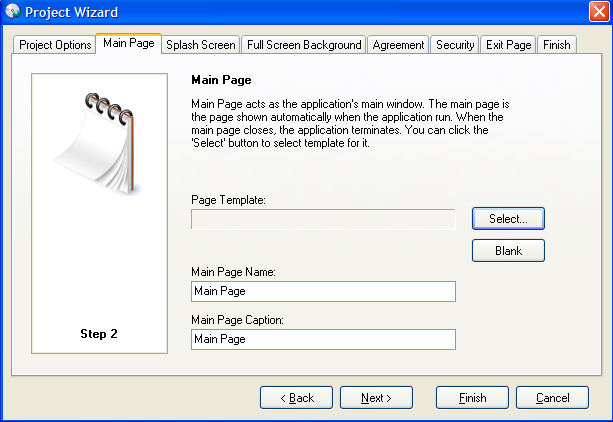
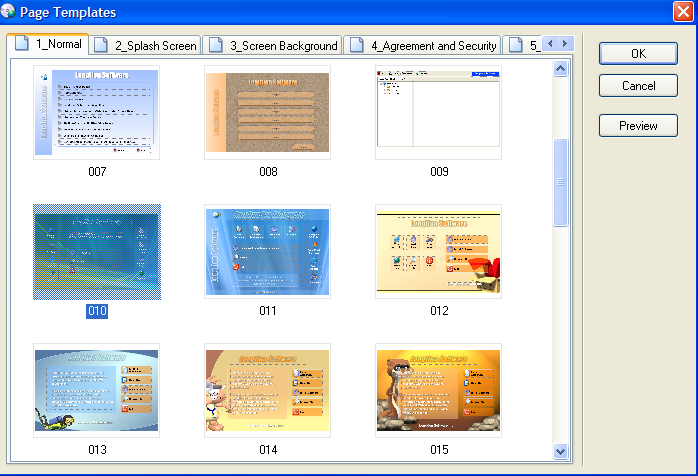


Figure – Page Templates



Template number 010 was used as the base template for PipeLine.

Next couple of screens of the wizard asks about whether to use splash screen, what kind of background graphics to use, whether to ask the user to agree to a user agreement legal statement, a security page to enter password to activate the product, an exit page to display to the user. None of these were used for the PipeLine project. Click finish to complete the project creation steps.

Customize the template

Customize the template number 010 that was used as the base. The process of customizing is basically:

* **Select a screen background image and color**. No background image was use. Color was set to white.
* **Set screen size**. Screen size was kept to default 697 X 480
* **Apply logo**. The USAID | DELIVER PROJECT logo was used. See screenshot below.
* **Create required pages**. After finalizing the page 1, a second page was created through Page > Duplicate Page menu option. The first page is meant to display program installations, links to website for further resources and a link to go to next screen. The second page displays the PipeLine documentation related menu choices.
* **Create menu options**. The template number 010 comes with default menu option choices. Those menu options were customized to build PipeLine related menu choices. The associated steps are illustrated below.

Apply logo

On the page 1 properties window, selecting the image item and select USAID logo. See illustration below:

Figure - Properties Window

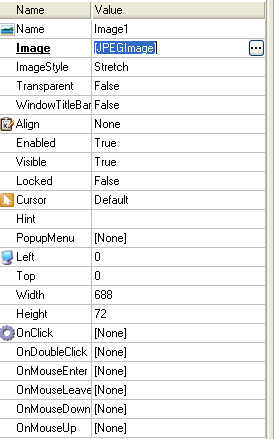
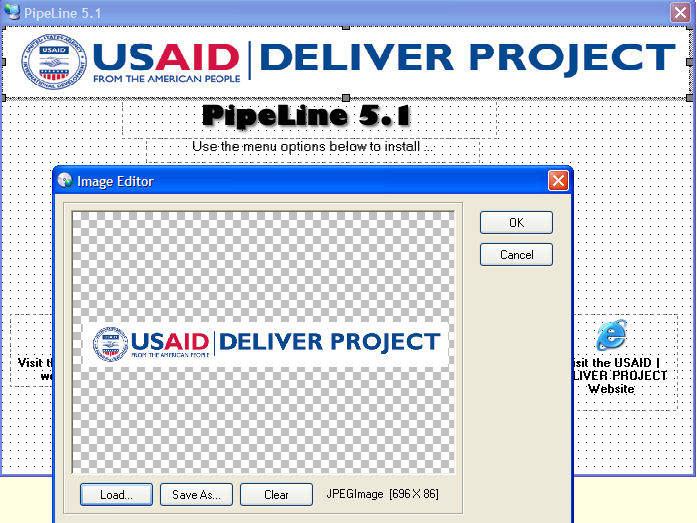


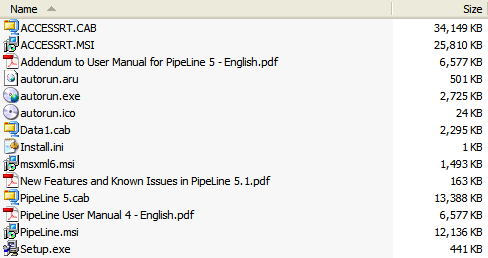
Figure - Image Editor



Place PipeLine media files in a folder

Create a folder to place all files related to PipeLine program. The AutoRun program allows creating folder and sub-folder structure for complex projects. However for PipeLine project, all files were kept on a single folder, as illustrated below.

Figure - PipeLine Files



Create menu choices

The template 010 comes with various menu choices. The following steps illustrate the process of building the PipeLine menu choices. Instead of repeating the steps for identical items, only the unique items were illustrated below.

Create menu options that calls the PipeLine installer to run

The menu options are defined through the properties window. The visual layout and the selected properties to define the PipeLine installation main menu option are illustrated below.

Figure - Visual Layout

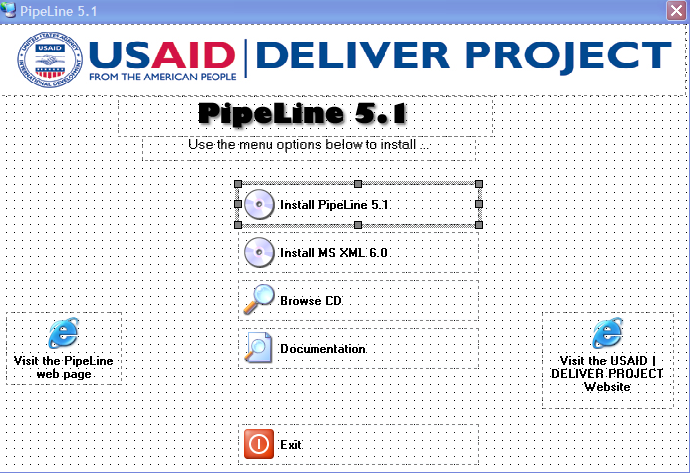
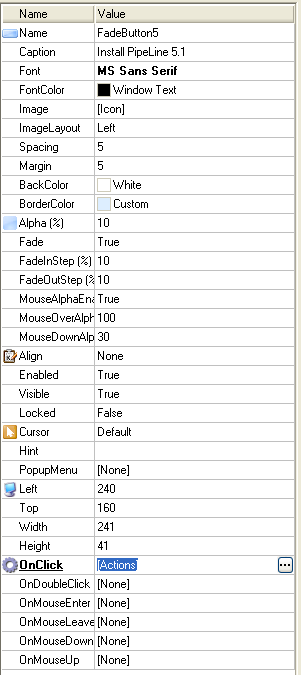
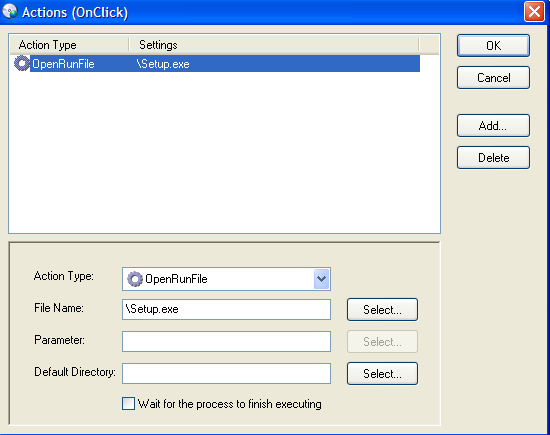


Figure - Main Menu Properties



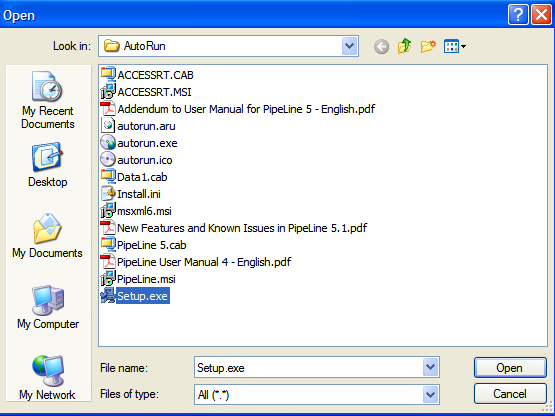
Through the properties window above, text, label, icon, formatting and onclick events were defined.

Figure - OnClick Actions



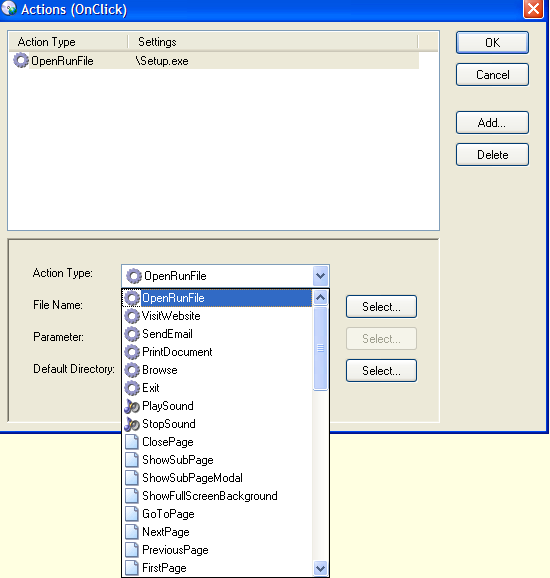
The onClick event executes the Setup.exe file placed at the root folder. Using the select button, select the Setup.exe file and click Open.

Figure - File Selection Window



Other available OnClick events are illustrated below

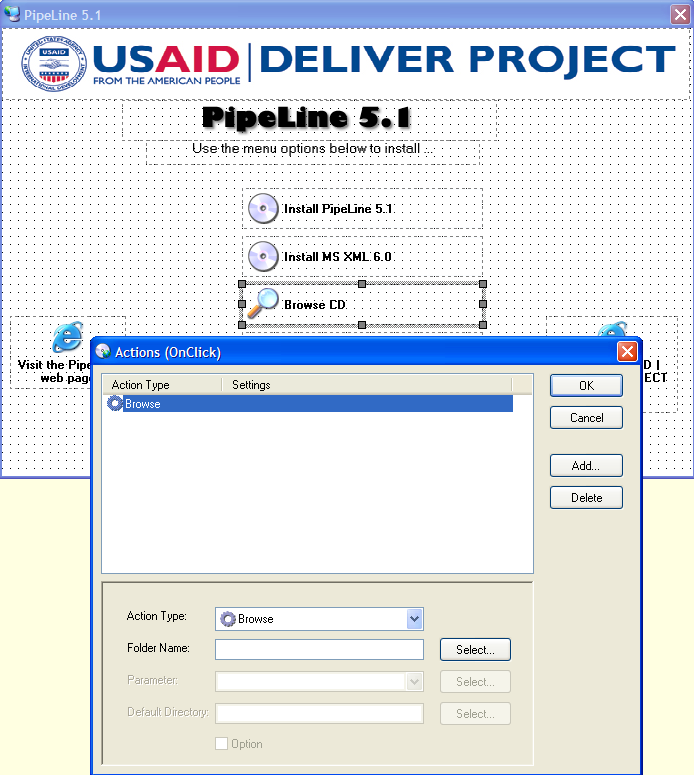
Figure – Available OnClick Events



Create CD browse menu option

Using Onclick > Browse event, the CD browse option was defined. See illustration below.

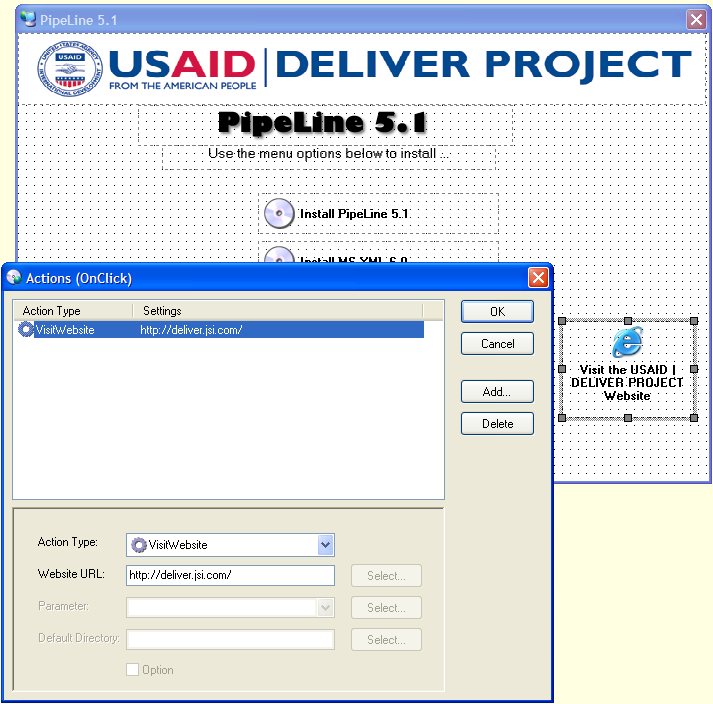
Figure – Browse CD OnClick Action



Create link to external website menu option

Using Onclick > VisitWebsite event, the navigation to the USAID | DELIVER PROJECT website and Visit PipeLine website links were created. This click will go to respective website, using the users’ default browser.

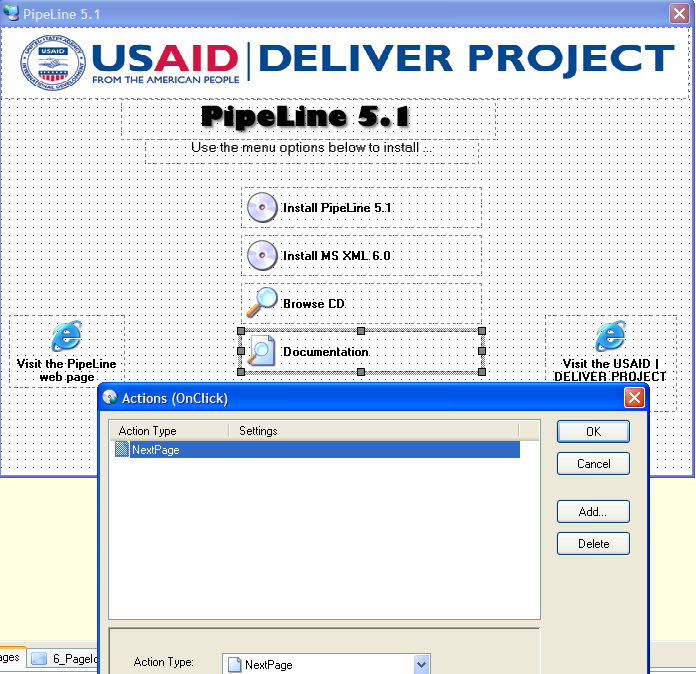
Figure - USAID|DELIVER PROJECT Website OnClick Action



Create link to next page menu option

Using Onclick > NextPage event, the navigation to next page was defined. See illustration below.

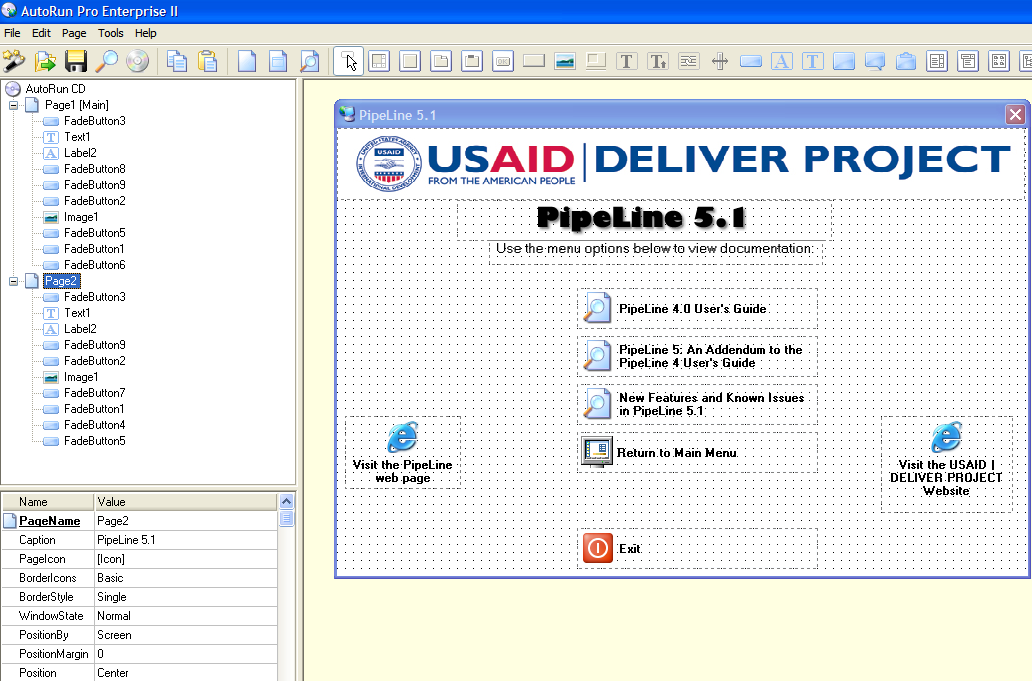
Figure - Documentation OnClick Action



Create second page

Using Page > Duplicate Page menu option second page was created.

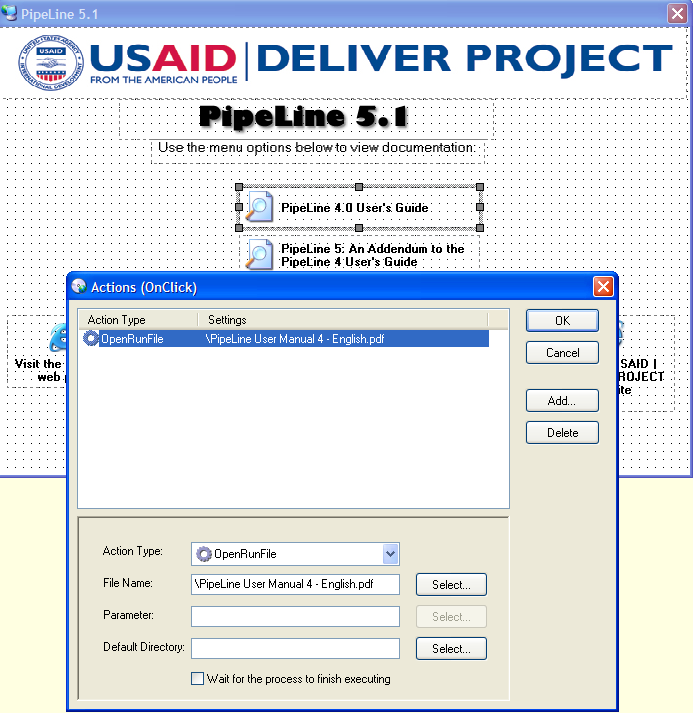
Figure - Duplicate Page



Create open PipeLine User Guide PDF file in Adobe Acrobat

Using OnClick > OpenRunFile click event, the open PDF file was defined.

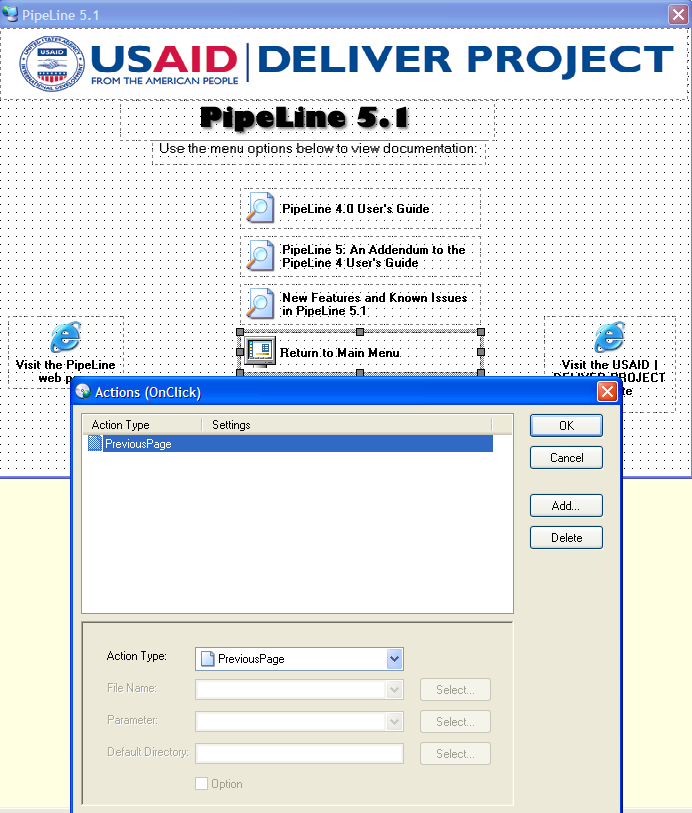
Figure - User Guide OnClick Action



Create Return to Main Menu

Using OnClick > PreviousPage click event, return to previous page was defined.

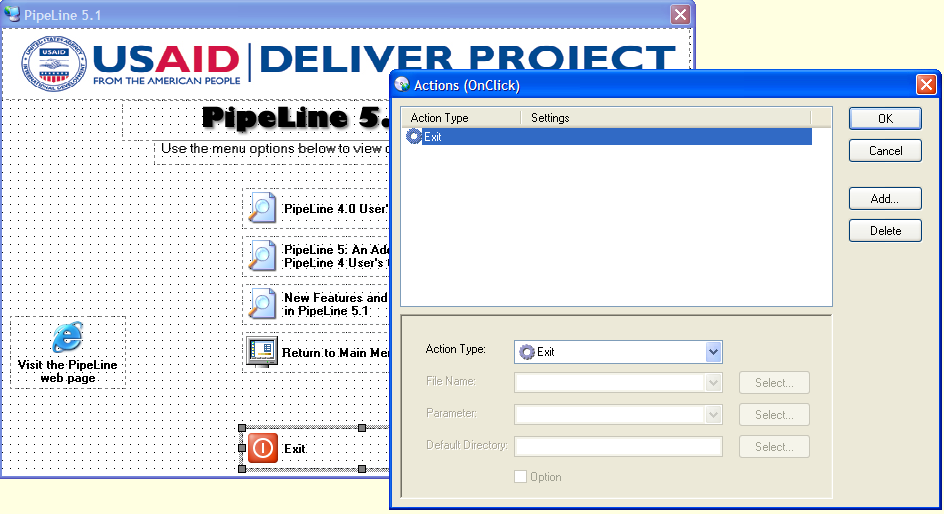
Figure - Return to Main Menu OnClick Action



Create Exit menu option

Using OnClick > Exit click event, exiting the CD Run program was defined.

Figure - Exit OnClick Action



Build the Executable AutoRun program and create the master CD

After the full two page menu choices were defined, it is time to test the project. Using File > Save and Test menu option, test the Auto install program. Once every element is tested and found acceptable, build the final version using File > Save and Publish menu option.

Figure - File Dropdown Menu

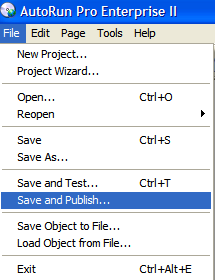


Figure - Publish Project

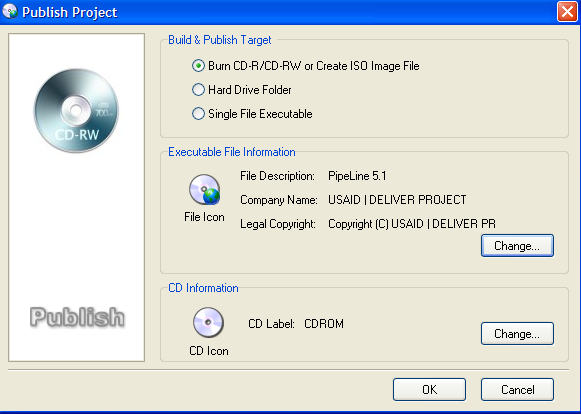
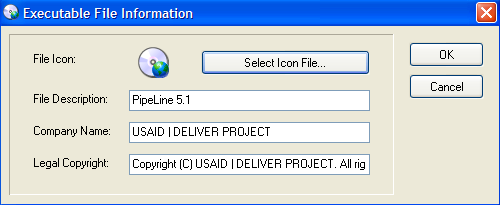


Figure - Executable File Information

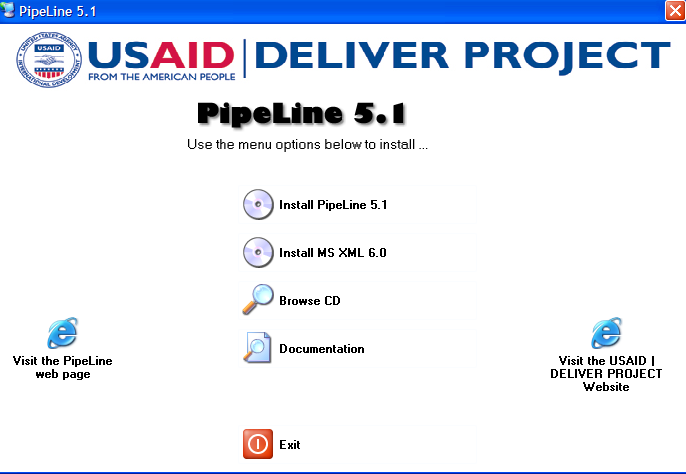


Test the final master CD and make required number of CD copies

After the master CD is build, test the CD under following operating environments:

* Windows XP
* Windows Vista
* Windows 7 32 bit
* Windows 7 64 bit

Figure - Final Auto-run CD



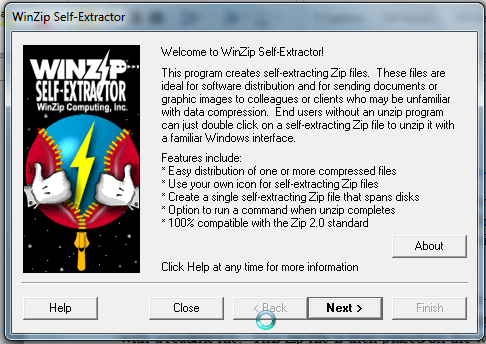
Final version of the Auto-run CD should look like above. After testing and acceptance, using a mass CD burner, publish required number of CDs. For professional quality CD label and high speed publishing, contact professional firms; provide master CD and related instructions.

Create Web Installer

Once the Auto run CD is created. The contents of the CD need to be packaged for distribution on the web. For this distribution, a self extracting exe file is created and zipped with a readme file. This zip file is then placed on the website for download. To create this self extracting exe file:

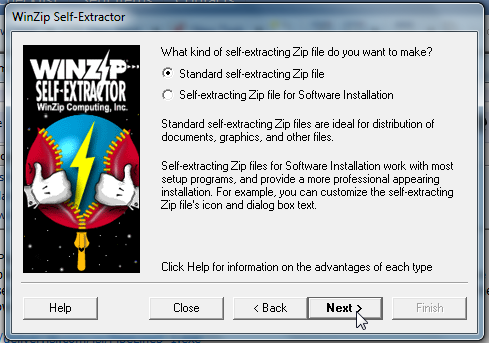
1. Using WinZip, zip all files found on the cd into a file PipeLine\_master.zip.
2. Add the password, 374949449384, to this zip file.
3. Open WinZip Self Extractor 2.2 and click next.

Figure - WinZip Self Extractor Welcome Window



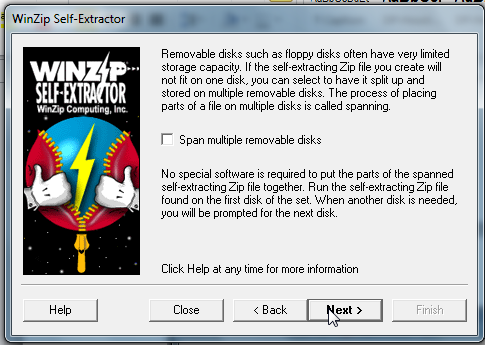
1. Select the option to create a standard self extracting zip file and click next.

Figure - WinZip Self Extractor Select Type Window



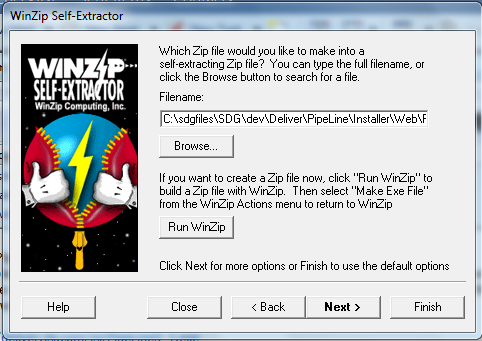
1. Uncheck the option to span multiple removable disks and click next.

Figure - WinZip Self Extractor Span Options Window



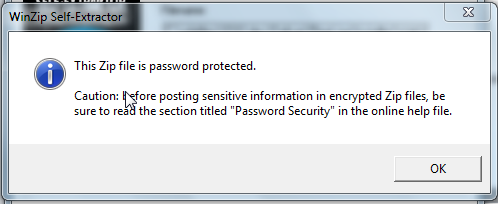
1. Click browse and choose the master zip file and click next.

Figure - WinZip Self Extractor File Selection Window



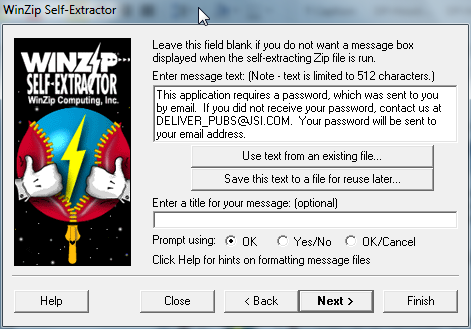
1. The following warning should appear and click OK.

Figure - WinZip Self Extractor Password Notification



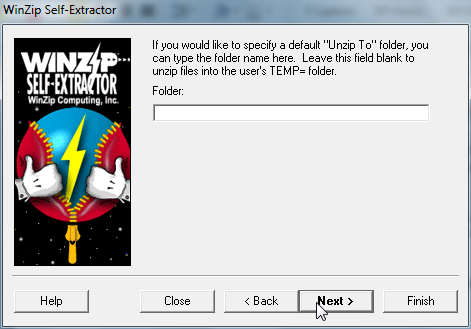
1. Click “Use test from an existing file” and select the password.txt file found in the installer folder and click next.

Figure - WinZip Self Extractor Message Text Option Window



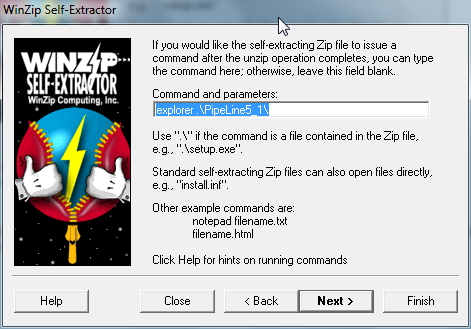
1. Leave folder field blank and click next.

Figure - WinZip Self Extractor "Unzip To" Option Window



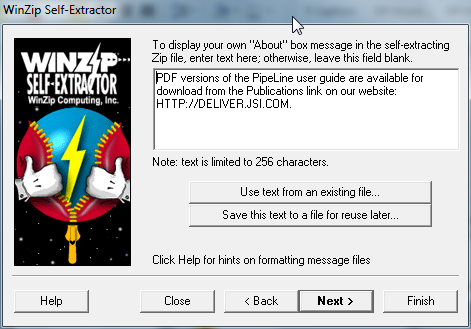
1. Enter value in command and parameters field as shown below and click next.

Figure - WinZip Self Extractor Command Options Window



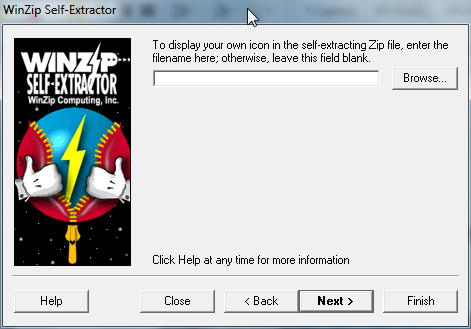
1. Click “Use test from an existing file” and select the about.txt file found in the installer folder and click next.

Figure - WinZip Self Extractor About Box Options Window



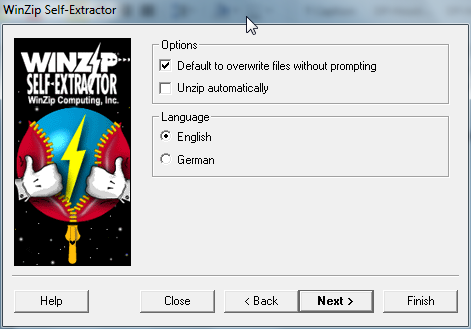
1. Leave the icon file name blank and click next.

Figure - WinZip Self Extractor Icon Selection Window



1. Choose the options specified below and click next.

Figure - WinZip Self Extractor Miscellaneous Options Window



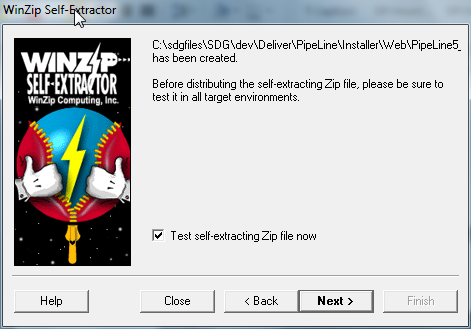
1. Click Next to create the exe file.

Figure - WinZip Self Extractor Ready to Create Window



1. When file is done being created, click next to text the file.

Figure - WinZip Self Extractor Test Window



1. After testing file, select No, I am Finished and click Exit close WinZip Self Extractor.
2. Go to installation folder and rename exe file to PipeLine5\_1.exe.
3. Select this file and the readme.txt file and create a new zip file named PipeLine5\_1.zip. This is the file to be distributed on the web.

Web-download

The full installer is downloadable from the SDG website (<http://sdg.jsi.com>). Upon downloading, user will getbe requested to fill out a information form. The password for unzipping the web installer will then be emailed to the address provided on this form.

Application Design

Overview

Access stores all database tables, queries, forms, reports, macros, and modules in the Access Jet database as a single file.

For query development, Access offers a "Query Designer", a graphical user interface that allows users to build queries without knowledge of the SQL programming language. In the Query Designer, users can "show" the data sources of the query (which can be tables or queries) and select the fields they want returned by clicking and dragging them into the grid. One can set up joins by clicking and dragging fields in tables to fields in other tables. Access allows users to view and manipulate the SQL code if desired. Any Access table, including linked tables from different data sources, can be used in a query.

When developing forms and reports that are linked to queries placing or moving items in the design view, Access runs the linked query in the background on any placement or movement of an item in that Form or Report. If the form or report is linked to a query that takes a long time to return records this means having to wait until the query has run before you can add/edit or move the next item in the form or report (this feature cannot be turned off).

Non-programmers can use the macro feature to automate simple tasks through a series of drop-down selections. Macros allow users to easily chain commands together such as running queries, importing or exporting data, opening and closing forms, previewing and printing reports, etc. Macros support basic logic (IF-conditions) and the ability to call other macros. Macros can also contain sub-macros which are similar to subroutines. Macros however, are limited in their functionality by a lack of programming loops and of advanced coding logic. PipeLine only uses macros for populating the custom menubar. This allows for ease in creating the menubar at runtime and in opening the proper dialog box.

The programming language available in Access is, as in other products of the Microsoft Office suite, Microsoft Visual Basic for Applications, which is nearly identical to Visual Basic 6.0 (VB6). VBA code can be stored in modules and code behind forms and reports. Modules can also be classes.

To manipulate data in tables and queries in VBA, Microsoft provides two database access libraries of COM components:

* Data Access Objects (DAO) (32-bit only), which is included in Access and Windows
* ActiveX Data Objects ActiveX Data Objects (ADO) (both 32-bit and 64-bit versions)

PipeLine uses DAO objects and so DAO must be a registered reference for the application.

Many Access developers use the Reddick naming convention, though this is not universal; it is a programming convention, not a DBMS-enforced rule. It is particularly helpful in VBA where references to object names may not indicate its data type (e.g. tbl for tables, qry for queries).

Split Database Architecture

Microsoft Access applications, like PipeLine, adopt a split-database architecture. The database is divided into a front-end database that contains the application objects (queries, forms, reports, macros, and modules), and is linked to tables stored in a back-end shared database containing the data. The "back-end" database can be stored in a location shared by many users, such as a file server. The "front-end" database is distributed to each user's desktop and linked to the shared database. Using this design, each user has a copy of Microsoft Access installed on their machine along with their application database. This reduces network traffic since the application is not retrieved for each use, and allows the front-end database to contain tables with data that is private to each user for storing settings or temporary data. This split-database design also allows development of the application independent of the data. When a new version is ready, the front-end database is replaced without impacting the data database.

Linked tables in Access use absolute paths rather than relative paths, so the development environment either has to have the same path as the production environment or a "dynamic-linker" routine can be written in VBA. In PipeLine, the links to the backend are stored in the ProgV4.mdb file. This also allows us to populate the Window menubar option.

This is not an economical setup across slow networks, or in large organizations separated by great distances, as it will result in excessive lag to database users. Therefore, when users are installing PipeLine, consideration needs to be made for their network speed.

Naming Conventions

Conventions used

Naming and coding conventions were not originally used. With version 3.0, these conventions have been used. All future programming should follow these guidelines.

The conventions that follow were developed to promote uniformity and consistency in naming various program modules. Variables and objects with familiar labels make source code easier to read.

All MSAccess databases should utilize Reddick VBA (RVBA), Version 6.01 (see Appendix C). Table summarizes the conventions used from the RVBA for database window objects.

Table 5 – Prefixes for Access Database Window Objects

|  |  |  |
| --- | --- | --- |
| Object | Prefix | Description |
| Table | tbl | Data Table |
|  | tmp | Temporary Table |
| Query | qsel | Select Query |
|  | qupd | Update Query |
|  | qdel | Delete Query |
|  | qapp | Append Query |
| Forms | frm | Form |
|  | fsub | Subform |
| Reports | rpt | Report |
|  | rsub | Subreport |
| Macros | mcr | Macro |
| Modules | bas | Module |

Table summarizes the conventions for object types found in form and reports.

Table 6 – Access Object Variable Prefixes

|  |  |
| --- | --- |
| Prefix | Object Type |
| chk | Checkbox |
| cbo | Combobox |
| cmd | Command Button |
| lbl | Label |
| lst | listbox |
| ole | ObjectFrame |
| opt | OptionButton |
| pge | Page |
| Tab | Tab Control |
| Txt | Text Box |
| Tgl | Toggle Button |

A prefix should first be added to each field in a table reflecting a two-digit abbreviation of the table name followed by an underscore and the descriptive naming convention for the object type. The first part of this prefix should reflect the table where it is located or the table name of the parent table if the field is a foreign key (i.e. ea\_datUpdated could be included in a table named tblExpectedActions and fr\_datReceived could be included in the same table thus showing its link to the parent table tblFundsReceived). Table summarizes the naming conventions for the object types in the tables.

Table 7 – Field Name Conventions

|  |  |
| --- | --- |
| Prefix | Object Type |
| bin | Binary |
| byt | Byte |
| lng | Long |
| cur | Currency |
| dat | Date/Time |
| dbl | Double |
| int | Integer |
| mem | Memo |
| sng | Single |
| str | Text |
| f | Yes/No |

Data Types

The following table contains the data types in the database tables along with other common data formats.

Table – Data Types

|  |  |  |  |
| --- | --- | --- | --- |
| Type  Field Size | Description | Format | Prefix |
| Yes/No | Used to setup fields containing boolean values. The default value is set to False. | True/False | f |
| Currency | Used to setup fields containing numeric values referring to US dollars. | 2 decimal places | cur |
| Number |  |  |  |
| Long Integer | Used to setup fields containing long integer values. |  | lng |
| Single | Used to setup fields containing single-precision floating-point values. |  | sng |
| Double | Used to setup fields containing double-precision floating-point values. |  | dbl |
| Byte | Used to setup fields containing byte values. |  | byt |
| Integer | Used to setup fields containing generic integer values. |  | int |
| Memo | Used to setup character fields spanning multiple lines. |  | mem |
| Date | Used to setup fields containing date values. | dd-mmm-yyyy | dat |
| Text  1 - 255 | Used to setup fields containing generic single-line text values. |  | txt |
| Zip Codes  10 | Used to setup fields containing zip codes. | 00000\-9999;0;\_ | txt |
| Phone #/Fax #  16/10 | Used to setup fields containing phone/fax numbers. | \(000”) “000\-0000\ a#####;1;\_ | txt |
| Percent | Used to setup fields holding percent values. | 2 decimal places | per |

Graphical User Interface

The following conventions were developed to promote uniformity and consistency in appearances. This MSAccess database utilizes Microsoft Application User Interface Guidelines created by JSI. (See Appendix E).

Explorer-Style Navigation

Resolution

The standard screen resolution has been updated to 1024x768. Please note that when applying this upgrade, the tree view, list view and detail view regions were increased proportionally.

Explorer-style navigation provides for easiest navigation of the application’s screens. It contains a treeview list of all screens available in the application, a list view of the records available to view and the detail view of the selected record. (For Reports the list view region will display the parameters and the detail view will display the report results based on selected parameters.) The following drawing defines the recognized screen regions and their sizes.

Figure – Sample Menu

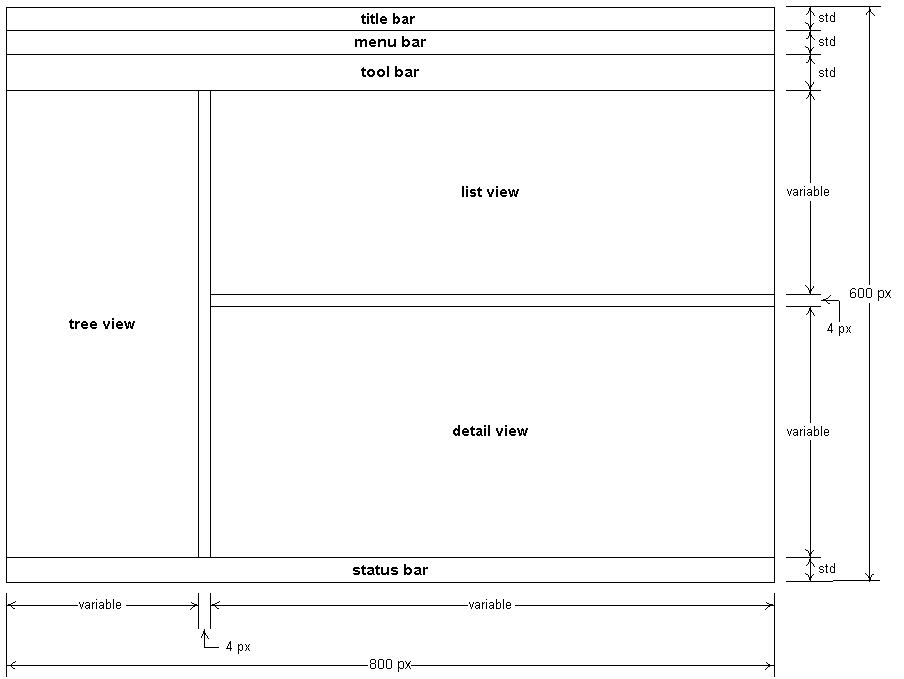


Table – Menu Element Specification Table

|  |  |
| --- | --- |
| Element | Comments |
| Title Bar | Should be set with the System Title and selected tree view title |
| Menu Bar | Should contain the minimum options needed for the form. |
| Tool Bar | (Optional) Should contain icons for quick access to various tools/features. |
| Tree View | On click should repopulate List View and Detail View with appropriate information. |
| List View | Should list the various record available in the detail view. In the case of reports, this should list the various parameters available for the report. |
| Detail View | Should display detailed information for the item selected in the list view or the parameters selected for the report. |
| Status Bar | Should provide quick simple explanation for the current field on the form. |

Menu Bar Specification

In PipeLine the Menu Bar contains the following options.

Table – Standard functionality of Switchboard Strip Menus

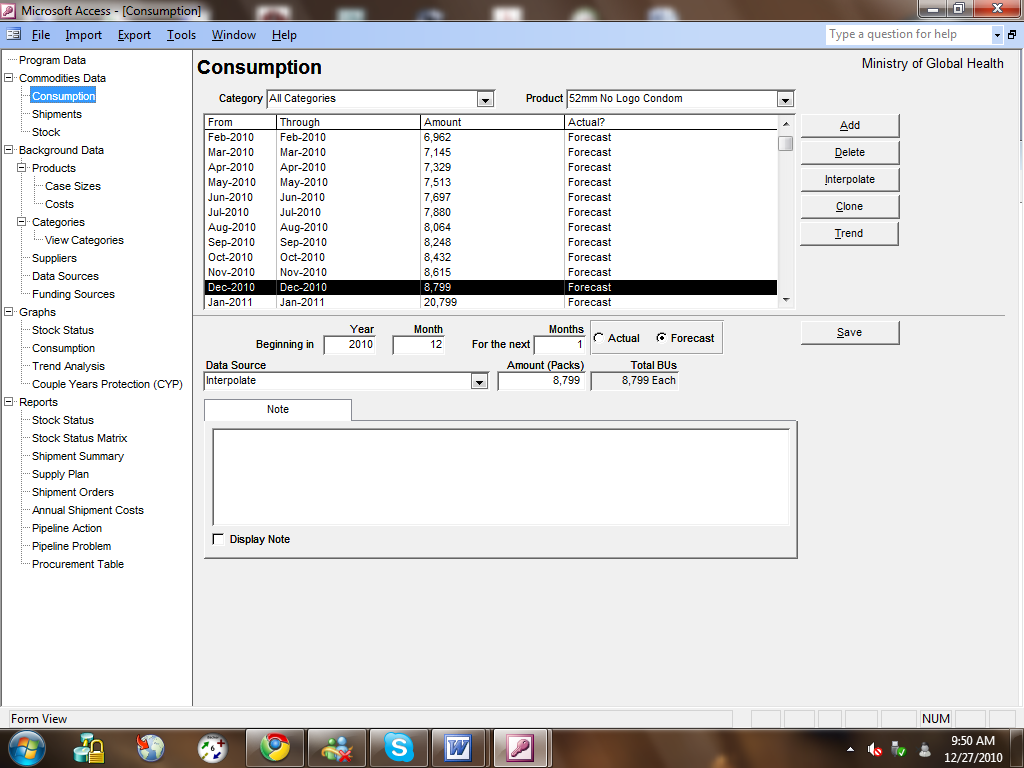
| Main Choice | Sub Choice | Description |
| --- | --- | --- |
| File | Exit | Close application and disconnect from the database |
| Import |  |  |
| Export |  |  |
| Tools |  |  |
| Window |  |  |
| Help | Help | Open online help for application. If no online help has been developed, should open Access help. |

Form Conventions

As shown in the figure below, the standard form lets users view record details and all updatable fields are enabled. Upon selecting an item in the list view, the form will filter to the correct record. (User may also select new to be taken to a new record where all fields are empty.)User may then click on SAVE to save the data. If the user modifies the data and tries to leave the form without clicking save, they will be prompted to save. To delete data from the application, the user will select the record in the list view and click the DELETE button. They will be prompted to verify the deletion.

The sample below depict standard for the list and detail screen.

Figure – Sample Detail Form



For each control in PipeLine a specific style must be applied. This is controlled by the tblStyle table in the application. The tlkTranslationText table contains the labels for all controls on all forms in the application. In this table, a style form the tblStyle is selected. When opening a form, the styles and labels are then applied to the control. This allows the entire application to be uniform and if the style requirements change, the developer only needs to modify the style table to apply the change to all forms. Below is the list of styles for each element:

Table - Element Styles

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Control Type | Font | Font Size | Fore Color | Font Weight | Underline? | Special Effect | Border Style | Back Color |
| Title | Arial | 14 |  | 700 | No | Flat | transparent |  |
| ProgramName | Arial | 10 |  | 400 | No | Flat | transparent |  |
| Main Selection | Arial | 10 |  | 400 | No | Sunken | solid |  |
| General Text | Arial | 8 |  | 400 | No | Sunken | solid |  |
| RptTitle1 | Arial | 12 |  | 700 | No | Flat | transparent |  |
| RptHeader | Arial | 8 |  | 400 | No | Flat | transparent |  |
| RptTitle2 | Arial | 10 |  | 400 | No | Flat | transparent |  |
| RptLabel | Arial | 8 |  | 400 | No | Flat | transparent |  |
| Cmd Button | arial | 8 |  | 400 | No |  |  |  |
| Checkbox |  |  |  |  | No | Sunken | solid |  |
| TabCtrl | Arial | 8 |  | 400 | No | Sunken |  |  |
| Frame |  |  |  |  | No | Sunken | solid |  |
| graph |  |  |  |  | No | Sunken | solid |  |
| Subform |  |  |  |  | No | Sunken | solid |  |
| RptGeneralHead w/o Border | arial | 8 |  | 400 | No | Flat | transparent |  |
| Hidden (fc=bc) | arial | 8 |  | 400 | No | Flat | solid |  |
| RptGroup1 | Arial | 8 |  | 700 | No | Flat | transparent |  |
| RptGroup2 | Arial | 8 |  | 400 | No | Flat | transparent |  |
| RptGroup3 | Arial | 8 |  | 400 | No | Flat | transparent |  |
| RptGroup4 | Arial | 8 |  | 400 | No | Flat | transparent |  |
| General Label | Arial | 8 |  | 400 | No | Flat | transparent |  |
| Main Select Label | Arial | 10 |  | 400 | No | Flat | transparent |  |
| General sfr Text | Arial | 8 |  | 400 | No | Flat | solid |  |
| RptGeneral w/Border | Arial | 8 |  | 400 | No | Flat | solid |  |
| RptGeneral w/o Border | Arial | 8 |  | 400 | No | Flat | transparent |  |
| RptGeneralHead w/Border | arial | 8 |  | 700 | No | Flat | solid |  |
| RptHidden (fc=bc) | arial | 8 |  | 400 | No | Flat | transparent | 10092543 |
| MoveButton | arial | 10 |  | 400 | No |  |  |  |
| Hypertext | Arial | 8 | 255 | 400 | No | Flat | transparent |  |
| RptTitle2\_Highlighted | Arial | 10 |  | 400 | No | Flat | transparent | 10092543 |
| RptGeneral w/Border Highlighted | Arial | 8 |  | 400 | No | Flat | solid | 10092543 |

Control Use

MSAccess features several different types of standard controls. It’s often possible to use two different controls to achieve the same general functionality. For example, a list of static values could be represented as a set of radio buttons or as pop list. The following guidelines should be used to determine what the best control for the job is.

Table – Control rules

|  |  |  |
| --- | --- | --- |
| Control | # | Rules |
| Check Boxes | 1 | Check Boxes should always be used for True/False fields unless the True/False paradigm does not obviously apply in which case two Radio Buttons should be used |
| Combo Box | 1 | Combo Boxes should be used in cases where a list may change or exceeds 8 values |
|  | 2 | The user should always select on a descriptive value rather than a code |
| List Box | 1 | List Boxes should be used in cases where a list may change and it’s important for the user to be able see the other options |
|  | 2 | The user should always select on a descriptive value rather than a code |
|  | 3 | List Boxes should be used in cases where it’s necessary for the user to keep track of multiple selected/ deselected options (i.e., Countries to include on a report) |
| Command Buttons | 1 | Command Buttons should be used to indicate a decision made by the user. |
| Radio Buttons | 1 | Radio Buttons should be used in cases where there are less than 8 choices and these choices are static. |
|  | 2 | Text Labels should be placed to right of Radio Buttons |
| Tab | 1 | Tab canvases should be used to model forms that would exceed a single screen. Also, tab canvases can be used to incorporate affiliated processes into a single module. |

Command Buttons

PipeLine contains thirteen standard command buttons that provide consistent functionality throughout the application. The buttons should only be used to provide the following functionality. Additional or different functionality should be provided by another command button.

When a command button is selected, the ON CLICK event procedure is fired, which calls the appropriate code. The table below describes the functionality of standard buttons.

Table – Functionality of standard buttons

| Button | Description |
| --- | --- |
| Add | Allow the user to enter a new record in the current recordset |
| Delete | Allow the user to delete the current record from the current recordset. |
| Save | Saves all changes to the current record since edit button was selected |
| Cancel | Cancels all changes to the current record since edit button was selected |
| Close | Closes form. Returns to calling form |
| Print | Sends the report directly to the printer. |
| Preview | Shows the print preview of the report. |
| PDF | Sends the report directly to the pdf printer specified. |
| ShowData | Shows data for report based on selected parameters in the detail view of the form. |
| Hide Data | Hides data for report in the detail view of the form. |

Report Conventions

Report Layout

The conventions on the following pages are to be used when designing and laying out reports:

Table – Report Checklist

| Group | Items | Specification |
| --- | --- | --- |
| General | Report Output | Should be horizontally centered on the page. |
|  | Default Paper Size (8.25 x 11) | Reports should be created to fit on both A4 and Letter sized paper. |
|  | Parameters | if parameters are required, they should be called from the list view section of the form so that the parameters can be validated. |
| Report Margins | Portrait Reports | .5" Top  .75 Bottom  .5" Left and Right |
|  | Landscape Reports | .5" Top and Bottom  .5" Left  .75" Right |
| Report headings | Report header including system title, report title, any other centered columns, run date, run time, page number | Set by tblStyle |
|  | 1st line of the report title | Report Name |
|  | 2nd line of the report title | SubReport Name |
|  | 3rd line of the report title | Description of Report including filtering summary (if applicable) |
|  | Report title spacing | Report titles should be stacked. There shouldn’t be a line or half line between titles. |
|  | Application Name, Report Display Name, and Program Name | Should be aligned with the left margin |
|  | Application Name | PipeLine 5.1 |
|  | Report Display Name | As specified on Program Form |
|  | Program Name | As specified on Program Form |
|  | Run Date, Run Time, and Page Number | Should be aligned with the right margin |
|  | Run Date | DD-MMM-YYYY Format |
|  | Run Time | Format as short time |
|  | Page Number | =”Page “ & [Page] & “ of “ & [Pages]) |
| Report Body | Report body text | Set by tblStyle |
|  | Report body labels | Set by tblStyle |
| Column Headings | Column headings | Justification of columns should match headings. Right justified amounts should have right justified headings, etc. |
|  | Column headings | Should appear on every page - but only once. |
| Totals | Totals | Should not appear by themselves on a page. |
|  | Total labels | Should appear to the left of the total row. The total label should be formatted bold and include the value of the group being totaled - i.e. Total Kenya or Total AVSC. Don’t use “Subtotal”. |

Icons

The standard MSAccess icons should be used.

VBA Coding Guidelines

VBA can be difficult to read when coded badly. PipeLine follows the Reddick VBA (RVBA) Coding Conventions (see Appendix ). The following outlines the proper way to document and structure VBA program units.

Procedure/Function Declarations

If an event procedure is more than 10 lines of code or is used more than once in the system, then a module should be created containing that public function. Any subroutine of the function should be included in the module and should be private if not being called by any other function.

Variable Declarations

Variables should also follow the naming conventions outlined in this chapter.

Comments

Every event procedure and functions should have a header comment. The header comment should include the following:

The purpose of the procedure or function

The purpose and definition of input parameters

The definition of any returned parameter

The date created and name of the primary developer

The dates, initials and technical notes of any subsequent developers

The header comments should appear in the following format.

‘ Comments:

‘ Parameters:

‘ Returns:

‘ Created:

‘ Modified:

‘––––––––––––––––––-

Any section of code that needs further explanation or is based on an external assumption should be commented. These comments should appear in the following format.

‘ This is the comment

White Space

Wherever the readability of the code will be enhanced (i.e. Around IF statements, FOR Loops, etc.) developer’s are encouraged to use white space.

Indentation

Developers should follow standard indentation practices when writing code.

Global Variables

Unless absolutely necessary, Global Variables should not be used. Instead use passed parameters or form level parameters

Database Schema

Overview

PipeLine has two distinct types of data: Commodity data and Background data. The Entity Relationship Diagram below shows the relations ship of the commodities data, i.e. Shipments, Inventory, and Consumption, and their relationship with the Product Background Table. The other background tables describe the Product Table and/or the commodity data: The second diagram below shows the relationships between these tables. The remaining tables are not part of the ERD since they either contain reference data, for example the translations tables, which are used to store information for the application itself, or are temporary tables used to store data temporarily when viewing forms and reports.

Entity Relationship Diagrams

Figure - ERD Diagram



Table and Column Descriptions

Table - List of Tables

| TableName | FieldName | DataType | FieldSize | Position | Primary Index | Secondary Index |
| --- | --- | --- | --- | --- | --- | --- |
| Action | ActionCode | Text | 10 | 1 | Yes | Yes (Duplicates OK) |
| ActionName | Text | 75 | 2 | No | No |
| CommodityCost\_Temp | ShipmentID | AutoNumber, Long Integer, Increment | 4 | 1 | No | No |
| ProductID | Text | 10 | 2 | No | No |
| SupplierID | Text | 10 | 3 | No | No |
| ShipDataSourceID | Text | 10 | 4 | No | No |
| ShipAmount | Number, Double | 8 | 5 | No | No |
| ShipPlannedDate | Date/Time | 8 | 6 | No | No |
| ShipOrderedDate | Date/Time | 8 | 7 | No | No |
| ShipShippedDate | Date/Time | 8 | 8 | No | No |
| ShipReceivedDate | Date/Time | 8 | 9 | No | No |
| ShipStatusCode | Text | 1 | 10 | No | No |
| ShipNote | Memo |  | 11 | No | No |
| ShipDateChanged | Date/Time | 8 | 12 | No | No |
| ShipFreightCost | Number, Single | 4 | 13 | No | No |
| ShipValue | Number, Double | 8 | 14 | No | No |
| ShipCaseLot | Number, Long Integer | 4 | 15 | No | No |
| ShipDisplayNote | Yes/No | 1 | 16 | No | No |
| ShipPO | Text | 50 | 17 | No | No |
| Old Shipment | Number, Double | 8 | 18 | No | No |
| ShipFundingSourceID | Text | 10 | 19 | No | No |
| UnitPrice | Text | 255 | 20 | No | No |
| Value | Number, Double | 8 | 21 | No | No |
| Freight | Number, Double | 8 | 22 | No | No |
| Cost | Number, Double | 8 | 23 | No | No |
| ProductName | Text | 50 | 24 | No | No |
| MethodName | Text | 100 | 25 | No | No |
| StatusName | Text | 50 | 26 | No | No |
| SortID | Text | 255 | 27 | No | No |
| CommodityPrice | ProductID | Text | 10 | 1 | Yes | Yes (Duplicates OK) |
| SupplierID | Text | 10 | 2 | Yes | Yes (Duplicates OK) |
| dtmEffective | Date/Time | 8 | 3 | Yes | Yes (Duplicates OK) |
| UnitPrice | Number, Single | 4 | 4 | No | No |
| dtmChanged | Date/Time | 8 | 5 | No | No |
| User | Text | 35 | 6 | No | No |
| Note | Text | 255 | 7 | No | No |
| fUserDefined | Yes/No | 1 | 8 | No | No |
| Consumption | ProductID | Text | 10 | 1 | Yes | Yes (Duplicates OK) |
| ConsStartYear | Number, Integer | 2 | 2 | Yes | Yes (Duplicates OK) |
| ConsStartMonth | Number, Integer | 2 | 3 | Yes | Yes (Duplicates OK) |
| ConsActualFlag | Yes/No | 1 | 4 | Yes | No |
| ConsNumMonths | Number, Byte | 1 | 5 | No | No |
| ConsAmount | Number, Double | 8 | 6 | No | No |
| ConsDataSourceID | Text | 10 | 7 | No | Yes (Duplicates OK) |
| ConsIflator | Number, Single | 4 | 8 | No | No |
| ConsNote | Memo |  | 9 | No | No |
| ConsDateChanged | Date/Time | 8 | 10 | No | No |
| ConsID | AutoNumber, Long Integer, Increment | 4 | 11 | No | Yes (No Duplicates) |
| ConsDisplayNote | Yes/No | 1 | 12 | No | Yes (Duplicates OK) |
| Old Consumption | Number, Double | 8 | 13 | No | No |
| Countries | Code | Text | 2 | 1 | Yes | Yes (Duplicates OK) |
| Country | Text | 50 | 2 | No | No |
| DataSource | DataSourceID | Text | 10 | 1 | Yes | Yes (Duplicates OK) |
| DataSourceName | Text | 50 | 2 | No | No |
| DataSourceTypeID | Text | 10 | 3 | No | Yes (Duplicates OK) |
| DataSourceType | DataSourceTypeID | Text | 10 | 1 | Yes | Yes (No Duplicates) |
| DataSourceTypeName | Text | 50 | 2 | No | Yes (No Duplicates) |
| FundingSource | FundingSourceID | Text | 10 | 1 | Yes | Yes (Duplicates OK) |
| FundingSourceName | Text | 50 | 2 | No | No |
| FundingNote | Memo |  | 3 | No | No |
| Inventory | ProductID | Text | 10 | 1 | No | Yes (Duplicates OK) |
| Period | Date/Time | 8 | 2 | No | Yes (Duplicates OK) |
| InvAmount | Number, Double | 8 | 3 | No | No |
| InvTransferFlag | Yes/No | 1 | 4 | No | No |
| InvNote | Memo |  | 5 | No | No |
| InvDateChanged | Date/Time | 8 | 6 | No | No |
| ctrIndex | AutoNumber, Long Integer, Increment | 4 | 7 | Yes | Yes (No Duplicates) |
| InvDisplayNote | Yes/No | 1 | 8 | No | No |
| InvDataSourceID | Text | 10 | 9 | No | Yes (Duplicates OK) |
| fImported | Yes/No | 1 | 10 | No | No |
| Old Inventory | Number, Double | 8 | 11 | No | No |
| Languages | LangTextID | Text | 3 | 1 | Yes | Yes (Duplicates OK) |
| LangIntegerID | Number, Integer | 2 | 2 | No | Yes (No Duplicates) |
| Language01 | Text | 25 | 3 | No | No |
| Language02 | Text | 25 | 4 | No | No |
| Language03 | Text | 25 | 5 | No | No |
| Language04 | Text | 25 | 6 | No | No |
| Language05 | Text | 25 | 7 | No | No |
| NativeText | Text | 50 | 8 | No | No |
| Method | MethodID | Text | 15 | 1 | Yes | Yes (Duplicates OK) |
| MethodName | Text | 100 | 2 | No | Yes (No Duplicates) |
| CYPFactor | Number, Single | 4 | 3 | No | No |
| MethodNote | Memo |  | 4 | No | No |
| ParentID | Number, Long Integer | 4 | 5 | No | Yes (Duplicates OK) |
| CategoryID | AutoNumber, Long Integer, Increment | 4 | 6 | No | Yes (No Duplicates) |
| fRollup | Yes/No | 1 | 7 | No | No |
| MonthlyAction | ProductID | Text | 10 | 1 | No | Yes (Duplicates OK) |
| ActionDate | Date/Time | 8 | 2 | No | Yes (Duplicates OK) |
| ActionCode | Text | 10 | 3 | No | Yes (Duplicates OK) |
| StatusCode | Text | 10 | 4 | No | Yes (Duplicates OK) |
| SupplierID | Text | 10 | 5 | No | Yes (Duplicates OK) |
| Amount | Number, Long Integer | 4 | 6 | No | No |
| ShipReceivedDate | Date/Time | 8 | 7 | No | No |
| ShipShippedDate | Date/Time | 8 | 8 | No | No |
| ShipOrderedDate | Date/Time | 8 | 9 | No | No |
| ShipPlannedDate | Date/Time | 8 | 10 | No | No |
| ShipmentID | Number, Long Integer | 4 | 11 | No | Yes (Duplicates OK) |
| MonthlyConsumption | ProductID | Text | 10 | 1 | Yes | Yes (Duplicates OK) |
| ConsYear | Number, Integer | 2 | 2 | Yes | Yes (Duplicates OK) |
| ConsMonth | Number, Byte | 1 | 3 | Yes | Yes (Duplicates OK) |
| ConsAmount | Number, Long Integer | 4 | 4 | No | No |
| ConsActualFlag | Yes/No | 1 | 5 | No | No |
| ConsDataSourceID | Text | 10 | 6 | No | Yes (Duplicates OK) |
| ConsPreData | Yes/No | 1 | 7 | No | Yes (Duplicates OK) |
| ConsBU | Number, Long Integer | 4 | 8 | No | No |
| MonthlyProblem | ProductID | Text | 10 | 1 | No | Yes (Duplicates OK) |
| ActionDate | Date/Time | 8 | 2 | No | Yes (Duplicates OK) |
| ProblemCode | Text | 10 | 3 | No | Yes (Duplicates OK) |
| StatusCode | Text | 10 | 4 | No | Yes (Duplicates OK) |
| SupplierID | Text | 10 | 5 | No | Yes (Duplicates OK) |
| Amount | Number, Long Integer | 4 | 6 | No | No |
| ShipReceivedDate | Date/Time | 8 | 7 | No | No |
| ShipShippedDate | Date/Time | 8 | 8 | No | No |
| ShipOrderedDate | Date/Time | 8 | 9 | No | No |
| ShipPlannedDate | Date/Time | 8 | 10 | No | No |
| ShipmentID | Text | 10 | 11 | No | Yes (Duplicates OK) |
| MonthlyShipment | ShipmentID | Number, Long Integer | 4 | 1 | Yes | No |
| ProductID | Text | 10 | 2 | No | Yes |
| SupplierID | Text | 10 | 3 | No | Yes (Duplicates OK) |
| ShipAmount | Number, Double | 8 | 4 | No | No |
| ShipStatusCode | Text | 1 | 5 | No | Yes (Duplicates OK) |
| ShipPlannedDate | Date/Time | 8 | 6 | No | No |
| ShipOrderedDate | Date/Time | 8 | 7 | No | No |
| ShipShippedDate | Date/Time | 8 | 8 | No | No |
| ShipReceivedDate | Date/Time | 8 | 9 | No | Yes |
| ShipDataSourceID | Text | 10 | 10 | No | Yes (Duplicates OK) |
| ShipBU | Number, Double | 8 | 11 | No | No |
| ShipFundingSourceID | Text | 50 | 12 | No | Yes (Duplicates OK) |
| MonthlyStock | ProductID | Text | 10 | 1 | Yes | Yes (Duplicates OK) |
| StockYear | Number, Integer | 2 | 2 | Yes | Yes (Duplicates OK) |
| StockMonth | Number, Byte | 1 | 3 | Yes | Yes (Duplicates OK) |
| ShipInMonth | Number, Integer | 2 | 4 | Yes | No |
| dtmRecord | Date/Time | 8 | 5 | No | No |
| StockShipAmount | Number, Long Integer | 4 | 6 | No | No |
| StockShipStatus | Text | 1 | 7 | No | No |
| StockShipSupplier | Text | 10 | 8 | No | No |
| StockConsAmount | Number, Long Integer | 4 | 9 | No | No |
| StockAdjustAmount | Number, Long Integer | 4 | 10 | No | No |
| StockInventoryAutoAdjust | Number, Long Integer | 4 | 11 | No | No |
| StockShipReceiveAmount | Number, Long Integer | 4 | 12 | No | No |
| StockBoMAmount | Number, Long Integer | 4 | 13 | No | No |
| StockAMCAmount | Number, Long Integer | 4 | 14 | No | No |
| StockShipPlanFlag | Yes/No | 1 | 15 | No | No |
| StockShipOrderFlag | Yes/No | 1 | 16 | No | No |
| StockShipShipFlag | Yes/No | 1 | 17 | No | No |
| ConsActualFlag | Yes/No | 1 | 18 | No | No |
| StockActualFlag | Yes/No | 1 | 19 | No | No |
| ShipinQTR | Number, Integer | 2 | 20 | No | No |
| MethodShipinMon | Number, Integer | 2 | 21 | No | No |
| MethodShipinQTR | Number, Integer | 2 | 22 | No | No |
| ProductBU | Number, Long Integer | 4 | 23 | No | No |
| StockInAmount | Number, Long Integer | 4 | 24 | No | No |
| StockOutAmount | Number, Long Integer | 4 | 25 | No | No |
| StockShipFundingSourceID | Text | 50 | 26 | No | Yes (Duplicates OK) |
| Period | Period\_Name | Text | 50 | 1 | No | Yes (No Duplicates) |
| Period\_Code | Number, Integer | 2 | 2 | Yes | No |
| Month | Yes/No | 1 | 3 | No | No |
| Beg\_Month | Text | 12 | 4 | No | No |
| End\_Month | Text | 12 | 5 | No | No |
| intMinMonths | Number, Long Integer | 4 | 6 | No | No |
| Period\_Name\_Arabic | Text | 12 | 7 | No | Yes (No Duplicates) |
| Beg\_Month\_Arabic | Text | 12 | 8 | No | No |
| End\_Month\_Arabic | Text | 12 | 9 | No | No |
| Period\_Name\_Sp | Text | 50 | 10 | No | Yes (No Duplicates) |
| Period\_Name\_Port | Text | 50 | 11 | No | Yes (No Duplicates) |
| Period\_Name\_Fra | Text | 50 | 12 | No | Yes (No Duplicates) |
| Period\_Name\_Eng | Text | 50 | 13 | No | Yes (No Duplicates) |
| Problem | ProblemCode | Text | 10 | 1 | Yes | Yes (No Duplicates) |
| ProblemName | Text | 50 | 2 | No | No |
| Product | ProductID | Text | 10 | 1 | Yes | Yes (Duplicates OK) |
| ProductName | Text | 50 | 2 | No | Yes (No Duplicates) |
| ProductMinMonths | Number, Byte | 1 | 3 | No | No |
| ProductMaxMonths | Number, Byte | 1 | 4 | No | No |
| SupplierID | Text | 10 | 5 | No | Yes (Duplicates OK) |
| MethodID | Text | 15 | 6 | No | Yes (Duplicates OK) |
| ProductActiveFlag | Yes/No | 1 | 7 | No | No |
| ProductActiveDate | Date/Time | 8 | 8 | No | No |
| DefaultCaseSize | Number, Long Integer | 4 | 9 | No | No |
| ProductNote | Memo |  | 10 | No | No |
| ProdCMax | Number, Byte | 1 | 11 | No | No |
| ProdCMin | Number, Byte | 1 | 12 | No | No |
| ProdDesStock | Number, Byte | 1 | 13 | No | No |
| txtInnovatorDrugName | Text | 50 | 14 | No | No |
| dblLowestUnitQty | Number, Double | 8 | 15 | No | No |
| txtLowestUnitMeasure | Text | 25 | 16 | No | No |
| txtSubstitutionList | Text | 255 | 17 | No | No |
| fPermittedInCountry | Yes/No | 1 | 18 | No | No |
| memAvailabilityNotes | Memo |  | 19 | No | No |
| fAvailabilityStatus | Yes/No | 1 | 20 | No | No |
| fUserDefined | Yes/No | 1 | 21 | No | No |
| strImportSource | Text | 10 | 22 | No | No |
| BUConversion | Number, Long Integer | 4 | 23 | No | No |
| txtPreferenceNotes | Text | 255 | 24 | No | No |
| lngAMCStart | Number, Long Integer | 4 | 25 | No | No |
| lngAMCMonths | Number, Long Integer | 4 | 26 | No | No |
| fAMCChanged | Yes/No | 1 | 27 | No | No |
| ProductFreightCost | ProductID | Text | 10 | 1 | Yes | Yes (Duplicates OK) |
| SupplierID | Text | 10 | 2 | Yes | Yes (Duplicates OK) |
| FreightCost | Number, Single | 4 | 3 | No | No |
| dtmChanged | Date/Time | 8 | 4 | No | No |
| ProductSupplierCaseSize | ProductID | Text | 10 | 1 | Yes | Yes (Duplicates OK) |
| SupplierID | Text | 10 | 2 | Yes | Yes (Duplicates OK) |
| dtmEffective | Date/Time | 8 | 3 | Yes | Yes (Duplicates OK) |
| intCaseSize | Number, Long Integer | 4 | 4 | No | No |
| dtmChanged | Date/Time | 8 | 5 | No | No |
| User | Text | 35 | 6 | No | No |
| Note | Text | 255 | 7 | No | No |
| Program | ProgramName | Text | 50 | 1 | Yes | Yes (Duplicates OK) |
| DataDirectory | Text | 250 | 2 | No | Yes (No Duplicates) |
| Language | Text | 3 | 3 | No | No |
| DefaultLeadTimePlan | Number, Single | 4 | 4 | No | No |
| DefaultLeadTimeOrder | Number, Single | 4 | 5 | No | No |
| DefaultLeadTimeShip | Number, Single | 4 | 6 | No | No |
| DefaultShipCost | Number, Single | 4 | 7 | No | No |
| ProgramContact | Text | 50 | 8 | No | No |
| Telephone | Text | 50 | 9 | No | No |
| Fax | Text | 50 | 10 | No | No |
| Email | Text | 50 | 11 | No | No |
| CountryCode | Text | 2 | 12 | No | Yes (Duplicates OK) |
| CountryName | Text | 50 | 13 | No | No |
| IsCurrent | Yes/No | 1 | 14 | No | No |
| Note | Memo |  | 15 | No | No |
| ProgramCode | Text | 12 | 16 | No | Yes (No Duplicates) |
| IsActive | Yes/No | 1 | 17 | No | No |
| StartSize | Text | 50 | 18 | No | No |
| IsDefault | Yes/No | 1 | 19 | No | No |
| fStartupFile | Yes/No | 1 | 20 | No | No |
| Shipment | ShipmentID | AutoNumber, Long Integer, Increment | 4 | 1 | Yes | Yes (No Duplicates) |
| ProductID | Text | 10 | 2 | No | Yes (Duplicates OK) |
| SupplierID | Text | 10 | 3 | No | Yes (Duplicates OK) |
| ShipDataSourceID | Text | 10 | 4 | No | Yes (Duplicates OK) |
| ShipAmount | Number, Double | 8 | 5 | No | No |
| ShipPlannedDate | Date/Time | 8 | 6 | No | No |
| ShipOrderedDate | Date/Time | 8 | 7 | No | No |
| ShipShippedDate | Date/Time | 8 | 8 | No | No |
| ShipReceivedDate | Date/Time | 8 | 9 | No | No |
| ShipStatusCode | Text | 1 | 10 | No | Yes (Duplicates OK) |
| ShipNote | Memo |  | 11 | No | No |
| ShipDateChanged | Date/Time | 8 | 12 | No | No |
| ShipFreightCost | Number, Single | 4 | 13 | No | Yes (Duplicates OK) |
| ShipValue | Number, Double | 8 | 14 | No | Yes (Duplicates OK) |
| ShipCaseLot | Number, Long Integer | 4 | 15 | No | No |
| ShipDisplayNote | Yes/No | 1 | 16 | No | Yes (Duplicates OK) |
| ShipPO | Text | 50 | 17 | No | Yes (Duplicates OK) |
| Old Shipment | Number, Double | 8 | 18 | No | No |
| ShipFundingSourceID | Text | 10 | 19 | No | Yes (Duplicates OK) |
| ShipSchedule | ProductID | Text | 10 | 1 | No | Yes (Duplicates OK) |
| ActionDate | Date/Time | 8 | 2 | No | Yes (Duplicates OK) |
| ActionCode | Text | 10 | 3 | No | Yes (Duplicates OK) |
| StatusCode | Text | 10 | 4 | No | Yes (Duplicates OK) |
| SupplierID | Text | 10 | 5 | No | Yes (Duplicates OK) |
| Amount | Number, Long Integer | 4 | 6 | No | No |
| ShipReceivedDate | Date/Time | 8 | 7 | No | No |
| ShipShippedDate | Date/Time | 8 | 8 | No | No |
| ShipOrderedDate | Date/Time | 8 | 9 | No | No |
| ShipPlannedDate | Date/Time | 8 | 10 | No | No |
| ShipmentID | Number, Long Integer | 4 | 11 | No | Yes (Duplicates OK) |
| Source | SupplierID | Text | 10 | 1 | Yes | Yes (No Duplicates) |
| SupplierName | Text | 50 | 2 | No | Yes (No Duplicates) |
| SupplierLeadTimePlan | Number, Single | 4 | 3 | No | No |
| SupplierLeadTimeOrder | Number, Single | 4 | 4 | No | No |
| SupplierLeadTimeShip | Number, Single | 4 | 5 | No | No |
| SupplierNote | Memo |  | 6 | No | No |
| Freight | Number, Single | 4 | 7 | No | No |
| DefaultSupplier | Yes/No | 1 | 8 | No | Yes (Duplicates OK) |
| Status | StatusCode | Text | 1 | 1 | Yes | Yes (Duplicates OK) |
| StatusName | Text | 50 | 2 | No | Yes (No Duplicates) |
| StatusOrder | Number, Integer | 2 | 3 | No | No |
| CableOrder | Number, Integer | 2 | 4 | No | No |
| Switchboard Items | SwitchboardID | Number, Long Integer | 4 | 1 | Yes | No |
| ItemNumber | Number, Integer | 2 | 2 | Yes | No |
| ItemText | Text | 255 | 3 | No | No |
| Command | Number, Integer | 2 | 4 | No | No |
| Argument | Text | 50 | 5 | No | No |
| TransCode | Text | 50 | 6 | No | Yes (Duplicates OK) |
| tblComments | cmt\_strProductID | Text | 10 | 1 | No | Yes (Duplicates OK) |
| cmt\_dtmPeriod | Date/Time | 8 | 2 | No | No |
| cmt\_strSource | Text | 50 | 3 | No | No |
| cmt\_memNote | Memo |  | 4 | No | No |
| tblDisplayBy | DisplayByID | Number, Long Integer | 4 | 1 | No | Yes (Duplicates OK) |
| DisplayByName | Text | 50 | 2 | No | Yes (No Duplicates) |
| tblEstimates | Month | Date/Time | 8 | 1 | Yes | No |
| intOrder | Number, Integer | 2 | 2 | No | Yes (Duplicates OK) |
| Estimated | Number, Double | 8 | 3 | No | No |
| hasActual | Yes/No | 1 | 4 | No | No |
| Actual | Number, Long Integer | 4 | 5 | No | No |
| Forecast | Number, Long Integer | 4 | 6 | No | No |
| LinEst | Number, Double | 8 | 7 | No | No |
| LinMin | Number, Double | 8 | 8 | No | No |
| LinMax | Number, Double | 8 | 9 | No | No |
| tblEstimatesForecast | Month | Date/Time | 8 | 1 | Yes | No |
| intOrder | Number, Integer | 2 | 2 | No | Yes (Duplicates OK) |
| Estimated | Number, Double | 8 | 3 | No | No |
| hasActual | Yes/No | 1 | 4 | No | No |
| Actual | Number, Long Integer | 4 | 5 | No | No |
| Forecast | Number, Long Integer | 4 | 6 | No | No |
| LinEst | Number, Double | 8 | 7 | No | No |
| LinMin | Number, Double | 8 | 8 | No | No |
| LinMax | Number, Double | 8 | 9 | No | No |
| ConsID | Number, Long Integer | 4 | 10 | No | No |
| fForecast | Yes/No | 1 | 11 | No | No |
| tblHelpContextID | FormName | Text | 255 | 1 | Yes | No |
| IsForm | Yes/No | 1 | 2 | Yes | No |
| Index | Number, Byte | 1 | 3 | Yes | No |
| EnglishContextID | Number, Long Integer | 4 | 4 | No | Yes (Duplicates OK) |
| FrenchContextID | Number, Long Integer | 4 | 5 | No | Yes (Duplicates OK) |
| SpanishContextID | Number, Long Integer | 4 | 6 | No | Yes (Duplicates OK) |
| Comment | Text | 255 | 7 | No | No |
| ArabicContextID | Number, Long Integer | 4 | 8 | No | Yes (Duplicates OK) |
| PortugueseContextID | Number, Long Integer | 4 | 9 | No | Yes (Duplicates OK) |
| tblImportProducts | strProductID | Text | 20 | 1 | No | Yes (Duplicates OK) |
| strName | Text | 100 | 2 | No | No |
| strDose | Text | 100 | 3 | No | No |
| lngCYP | Number, Double | 8 | 4 | No | No |
| dtmExport | Date/Time | 8 | 5 | No | No |
| fProcessed | Yes/No | 1 | 6 | No | No |
| lngID | AutoNumber, Long Integer, Increment | 4 | 7 | Yes | Yes (Duplicates OK) |
| strSource | Text | 255 | 8 | No | No |
| strMapping | Text | 50 | 9 | No | No |
| tblImportProductsSCMS | strProductID | Text | 20 | 1 | No | Yes (Duplicates OK) |
| strName | Text | 255 | 2 | No | No |
| strDose | Text | 100 | 3 | No | No |
| lngCYP | Number, Double | 8 | 4 | No | No |
| dtmExport | Date/Time | 8 | 5 | No | No |
| fProcessed | Yes/No | 1 | 6 | No | No |
| lngID | AutoNumber, Long Integer, Increment | 4 | 7 | Yes | Yes (Duplicates OK) |
| strSource | Text | 255 | 8 | No | No |
| strMapping | Text | 255 | 9 | No | No |
| strMappingFull | Text | 255 | 10 | No | No |
| strPLMapping | Text | 255 | 11 | No | No |
| strPLMappingFull | Text | 255 | 12 | No | No |
| fSelect | Yes/No | 1 | 13 | No | No |
| fLocked | Yes/No | 1 | 14 | No | No |
| fNoBlanks | Yes/No | 1 | 15 | No | No |
| fUserDefined | Yes/No | 1 | 16 | No | No |
| strProductGroup | Text | 50 | 17 | No | No |
| strInnovatorName | Text | 50 | 18 | No | No |
| dblLowestUnitQty | Number, Long Integer | 4 | 19 | No | No |
| strLowestUnitMeasure | Text | 50 | 20 | No | No |
| intQuantificationFactor | Number, Double | 8 | 21 | No | No |
| strSourceName | Text | 50 | 22 | No | No |
| strSystemName | Text | 50 | 23 | No | No |
| strShortName | Text | 50 | 24 | No | No |
| fPLLocked | Yes/No | 1 | 25 | No | No |
| tblImportRecords | strProductID | Text | 50 | 1 | No | Yes (Duplicates OK) |
| dtmPeriod | Date/Time | 8 | 2 | No | No |
| lngConsumption | Number, Long Integer | 4 | 3 | No | No |
| lngAdjustment | Number, Long Integer | 4 | 4 | No | No |
| dblDataInterval | Number, Double | 8 | 5 | No | No |
| lngParentID | Number, Long Integer | 4 | 6 | No | Yes (Duplicates OK) |
| tblImportRecordsSCMS | strProductID | Text | 50 | 1 | No | Yes (Duplicates OK) |
| dtmPeriod | Date/Time | 8 | 2 | No | No |
| lngConsumption | Number, Double | 8 | 3 | No | No |
| lngAdjustment | Number, Long Integer | 4 | 4 | No | No |
| dblDataInterval | Number, Double | 8 | 5 | No | No |
| lngParentID | Number, Long Integer | 4 | 6 | No | Yes (Duplicates OK) |
| tblInterpolate | Month | Date/Time | 8 | 1 | Yes | No |
| intOrder | Number, Integer | 2 | 2 | No | Yes (Duplicates OK) |
| intercept | Number, Double | 8 | 3 | No | No |
| slope | Number, Double | 8 | 4 | No | No |
| Estimated | Number, Double | 8 | 5 | No | No |
| fForecast | Yes/No | 1 | 6 | No | No |
| ConsID | Number, Long Integer | 4 | 7 | No | No |
| tblMakeGraph\_Temp | ProductID | Text | 10 | 1 | No | Yes (Duplicates OK) |
| Date | Date/Time | 8 | 2 | No | No |
| Actual | Number, Long Integer | 4 | 3 | No | No |
| Forecast | Number, Long Integer | 4 | 4 | No | No |
| ConsAmount | Number, Long Integer | 4 | 5 | No | No |
| ConsActualFlag | Yes/No | 1 | 6 | No | No |
| tblStatusbartext | FormName | Text | 255 | 1 | Yes | No |
| ControlName | Text | 255 | 2 | Yes | No |
| Index | Number, Byte | 1 | 3 | Yes | No |
| EnglishStatus | Text | 255 | 4 | No | No |
| FrenchStatus | Text | 255 | 5 | No | No |
| SpanishStatus | Text | 255 | 6 | No | No |
| Comment | Text | 255 | 7 | No | No |
| property | Number, Long Integer | 4 | 8 | No | No |
| ID | Text | 50 | 9 | No | Yes (Duplicates OK) |
| tblStyle | lngType | AutoNumber, Long Integer, Increment | 4 | 1 | Yes | No |
| txtTypeName | Text | 50 | 2 | No | No |
| txtFont | Text | 50 | 3 | No | No |
| lngFontSize | Number, Long Integer | 4 | 4 | No | No |
| txtForeColor | Text | 50 | 5 | No | No |
| txtFontweight | Text | 50 | 6 | No | No |
| fUnderline | Yes/No | 1 | 7 | No | No |
| lngSpecialEffect | Number, Long Integer | 4 | 8 | No | No |
| lngBorderStyle | Number, Long Integer | 4 | 9 | No | No |
| txtBackColor | Text | 50 | 10 | No | No |
| tblSysParameters | strParmName | Text | 50 | 1 | Yes | No |
| strParmValue | Text | 50 | 2 | No | No |
| strParmDesc | Text | 255 | 3 | No | No |
| tblTempCPT | r\_code | Text | 12 | 1 | Yes | Yes (Duplicates OK) |
| p\_code | Text | 10 | 2 | Yes | Yes (Duplicates OK) |
| tb\_yr | Number, Integer | 2 | 3 | No | No |
| tb\_Prep | Date/Time | 8 | 4 | No | No |
| tb\_who | Text | 50 | 5 | No | No |
| tb\_min\_lvl1 | Number, Integer | 2 | 6 | No | No |
| tb\_min\_lvl2 | Number, Integer | 2 | 7 | No | No |
| tb\_m\_eoy1 | Number, Integer | 2 | 8 | No | No |
| tb\_m\_eoy2 | Number, Integer | 2 | 9 | No | No |
| tb\_begQty | Number, Double | 8 | 10 | No | No |
| tb\_est1 | Number, Single | 4 | 11 | No | No |
| tb\_est2 | Number, Single | 4 | 12 | No | No |
| tb\_est3 | Number, Single | 4 | 13 | No | No |
| tb\_est4 | Number, Single | 4 | 14 | No | No |
| tb\_est5 | Number, Single | 4 | 15 | No | No |
| tb\_lost1 | Number, Single | 4 | 16 | No | No |
| tb\_lost2 | Number, Single | 4 | 17 | No | No |
| tb\_lost3 | Number, Single | 4 | 18 | No | No |
| tb\_lost4 | Number, Single | 4 | 19 | No | No |
| tb\_lost5 | Number, Single | 4 | 20 | No | No |
| tb\_AidIn1 | Number, Single | 4 | 21 | No | No |
| tb\_AidIn2 | Number, Single | 4 | 22 | No | No |
| tb\_AidIn3 | Number, Single | 4 | 23 | No | No |
| tb\_AidDue2 | Number, Single | 4 | 24 | No | No |
| tb\_AidDue3 | Number, Single | 4 | 25 | No | No |
| tb\_AidDue4 | Number, Single | 4 | 26 | No | No |
| tb\_AidDue5 | Number, Single | 4 | 27 | No | No |
| tb\_des3 | Number, Single | 4 | 28 | No | No |
| tb\_des4 | Number, Single | 4 | 29 | No | No |
| tb\_des5 | Number, Single | 4 | 30 | No | No |
| tb\_Ord3 | Number, Single | 4 | 31 | No | No |
| tb\_Ord4 | Number, Single | 4 | 32 | No | No |
| tb\_Ord5 | Number, Single | 4 | 33 | No | No |
| tb\_TransIn1 | Number, Single | 4 | 34 | No | No |
| tb\_TransIn2 | Number, Single | 4 | 35 | No | No |
| tb\_TransIn3 | Number, Single | 4 | 36 | No | No |
| tb\_TransIn4 | Number, Single | 4 | 37 | No | No |
| tb\_TransIn5 | Number, Single | 4 | 38 | No | No |
| tb\_TransOut1 | Number, Single | 4 | 39 | No | No |
| tb\_TransOut2 | Number, Single | 4 | 40 | No | No |
| tb\_TransOut3 | Number, Single | 4 | 41 | No | No |
| tb\_TransOut4 | Number, Single | 4 | 42 | No | No |
| tb\_TransOut5 | Number, Single | 4 | 43 | No | No |
| tb\_Max\_lvl1 | Number, Integer | 2 | 44 | No | No |
| tb\_Max\_lvl2 | Number, Integer | 2 | 45 | No | No |
| tblTempCPTProducts | ProductID | Text | 10 | 1 | Yes | No |
| tblTempCPTShipments | ShipmentID | Number, Long Integer | 4 | 1 | Yes | No |
| ProductID | Text | 10 | 2 | No | Yes |
| SupplierID | Text | 10 | 3 | No | Yes (Duplicates OK) |
| ShipAmount | Number, Double | 8 | 4 | No | No |
| ShipStatusCode | Text | 1 | 5 | No | Yes (Duplicates OK) |
| ShipPlannedDate | Date/Time | 8 | 6 | No | No |
| ShipOrderedDate | Date/Time | 8 | 7 | No | No |
| ShipShippedDate | Date/Time | 8 | 8 | No | No |
| ShipReceivedDate | Date/Time | 8 | 9 | No | Yes |
| ShipDataSourceID | Text | 10 | 10 | No | Yes (Duplicates OK) |
| ShipBU | Number, Double | 8 | 11 | No | No |
| tblTempImportShipments | Recipient | Text | 255 | 1 | No | Yes (Duplicates OK) |
| ProductID | Text | 255 | 2 | No | Yes (Duplicates OK) |
| ReceiptDate | Date/Time | 8 | 3 | No | No |
| Quantity | Number, Double | 8 | 4 | No | No |
| StatusCode | Text | 255 | 5 | No | Yes (Duplicates OK) |
| newvernID | Text | 255 | 6 | No | Yes (Duplicates OK) |
| Comment | Text | 255 | 7 | No | No |
| tblTempNewvernShipments | Recipient | Text | 255 | 1 | No | Yes (Duplicates OK) |
| ProductID | Text | 255 | 2 | No | Yes (Duplicates OK) |
| TB\_YR | Number, Integer | 2 | 3 | No | No |
| ReceiptDate | Date/Time | 8 | 4 | No | No |
| Quantity | Number, Double | 8 | 5 | No | No |
| StatusCode | Text | 255 | 6 | No | Yes (Duplicates OK) |
| NewvernID | Text | 255 | 7 | No | Yes (Duplicates OK) |
| Comment | Text | 255 | 8 | No | No |
| tempForeCons | ProductID | Text | 10 | 1 | No | No |
| ConsYear | Number, Integer | 2 | 2 | Yes | No |
| ConsMonth | Number, Integer | 2 | 3 | Yes | No |
| ConsAmount | Number, Long Integer | 4 | 4 | No | No |
| ConsDataSourceID | Text | 10 | 5 | No | No |
| tempXtabStockStatusMatrix | KeyID | Text | 10 | 1 | No | No |
| KeyName | Text | 50 | 2 | No | No |
| Unit | Text | 25 | 3 | No | No |
| StockMin | Number, Byte | 1 | 4 | No | No |
| StockMax | Number, Byte | 1 | 5 | No | No |
| MOS | Number, Double | 8 | 6 | No | No |
| StockDate | Date/Time | 8 | 7 | No | No |
| tlkReportExport | ProductID | Text | 10 | 1 | No | No |
| txtMonYear | Date/Time | 8 | 2 | No | No |
| txtStockBoMAmount | Number, Long Integer | 4 | 3 | No | No |
| StockShipAmount | Number, Long Integer | 4 | 4 | No | No |
| txtStockShipStatus | Text | 255 | 5 | No | No |
| txtStockShipSupplier | Text | 255 | 6 | No | No |
| txtActCons | Number, Long Integer | 4 | 7 | No | No |
| Actual | Text | 255 | 8 | No | No |
| TotalAdjustAmount | Number, Long Integer | 4 | 9 | No | No |
| txtStockinMonths | Number, Double | 8 | 10 | No | No |
| txttoMax | Number, Double | 8 | 11 | No | No |
| txtShortSurp | Number, Double | 8 | 12 | No | No |
| txtEoMAmount | Number, Long Integer | 4 | 13 | No | No |
| tlkTranslationText | DocumentType | Number, Long Integer | 4 | 1 | Yes | No |
| FormName | Text | 255 | 2 | Yes | No |
| ControlName | Text | 255 | 3 | Yes | No |
| Property | Number, Long Integer | 4 | 4 | Yes | No |
| Index | Number, Byte | 1 | 5 | Yes | No |
| TranslationID | Text | 255 | 6 | No | Yes (Duplicates OK) |
| StyleID | Number, Long Integer | 4 | 7 | No | Yes (Duplicates OK) |
| txtJustification | Number, Long Integer | 4 | 8 | No | No |
| FrenchStatus | Text | 255 | 9 | No | No |
| SpanishStatus | Text | 255 | 10 | No | No |
| EnglishStatus | Text | 255 | 11 | No | No |
| NewRecord | Yes/No | 1 | 12 | No | No |
| ArabicStatus | Text | 255 | 13 | No | No |
| bytReadingOrderA | Number, Byte | 1 | 14 | No | No |
| PortugueseStatus | Text | 255 | 15 | No | No |
| comment | Text | 50 | 16 | No | No |
| tlkTreeview | lngID | AutoNumber, Long Integer, Increment | 4 | 1 | Yes | Yes (Duplicates OK) |
| dblParent | Number, Double | 8 | 2 | No | Yes |
| dblsortorder | Number, Double | 8 | 3 | No | Yes |
| txtName | Text | 50 | 4 | No | No |
| txtformname | Text | 50 | 5 | No | No |
| txtSpanish | Text | 50 | 6 | No | No |
| txtFrench | Text | 50 | 7 | No | No |
| NewRecord | Yes/No | 1 | 8 | No | No |
| TranslationID | Text | 255 | 9 | No | Yes (Duplicates OK) |
| txtArabic | Text | 50 | 10 | No | No |
| txtPortuguese | Text | 50 | 11 | No | No |
| tlkUpgrade | id | AutoNumber, Long Integer, Increment | 4 | 1 | Yes | Yes (Duplicates OK) |
| Type | Text | 50 | 2 | No | No |
| strSQL1 | Memo |  | 3 | No | No |
| strSQL2 | Memo |  | 4 | No | No |
| strSQL3 | Memo |  | 5 | No | No |
| strTable | Text | 50 | 6 | No | No |
| version | Number, Long Integer | 4 | 7 | No | No |
| lngOrder | Number, Long Integer | 4 | 8 | No | No |
| tlkYears | lngYear | Number, Long Integer | 4 | 1 | No | No |
| tmpCaseSizes | ProductID | Text | 10 | 1 | No | Yes (Duplicates OK) |
| SupplierID | Text | 10 | 2 | No | Yes (Duplicates OK) |
| dtmEffective | Date/Time | 8 | 3 | No | Yes (Duplicates OK) |
| intCaseSize | Number, Long Integer | 4 | 4 | No | No |
| dtmChanged | Date/Time | 8 | 5 | No | No |
| User | Text | 35 | 6 | No | No |
| Note | Text | 255 | 7 | No | No |
| tmpCategory | MethodID | Text | 10 | 1 | No | Yes (Duplicates OK) |
| MethodName | Text | 50 | 2 | No | No |
| CYPFactor | Number, Single | 4 | 3 | No | No |
| MethodNote | Memo |  | 4 | No | No |
| fRollup | Yes/No | 1 | 5 | No | No |
| ParentID | Text | 255 | 6 | No | Yes (Duplicates OK) |
| tmpCategoryProduct | MethodID | Text | 10 | 1 | No | Yes (Duplicates OK) |
| ProductID | Text | 10 | 2 | No | Yes (Duplicates OK) |
| tmpCloneConsumption | ProductID | Text | 10 | 1 | No | Yes (Duplicates OK) |
| ConsStartYear | Number, Integer | 2 | 2 | No | Yes (Duplicates OK) |
| ConsStartMonth | Number, Integer | 2 | 3 | No | Yes (Duplicates OK) |
| ConsActualFlag | Yes/No | 1 | 4 | No | No |
| ConsNumMonths | Number, Byte | 1 | 5 | No | No |
| ConsAmount | Number, Double | 8 | 6 | No | No |
| ConsDataSourceID | Text | 10 | 7 | No | Yes (Duplicates OK) |
| ConsIflator | Number, Single | 4 | 8 | No | No |
| ConsNote | Memo |  | 9 | No | No |
| ConsDateChanged | Date/Time | 8 | 10 | No | No |
| ConsDisplayNote | Yes/No | 1 | 11 | No | Yes (Duplicates OK) |
| ConsAmountBU | Number, Double | 8 | 12 | No | No |
| ConsClonedAmountBU | Number, Double | 8 | 13 | No | No |
| ConsClonedProductID | Text | 10 | 14 | No | Yes (Duplicates OK) |
| ConsfDataAccepted | Yes/No | 1 | 15 | No | No |
| ConsfConflict | Yes/No | 1 | 16 | No | No |
| tmpCloneProducts | ProductID | Text | 10 | 1 | No | Yes (Duplicates OK) |
| ProductName | Text | 50 | 2 | No | No |
| dblLowestUnitQty | Number, Double | 8 | 3 | No | No |
| fSelected | Number, Long Integer | 4 | 4 | No | No |
| dblFactor | Number, Double | 8 | 5 | No | No |
| tmpConsumption | ProductID | Text | 10 | 1 | No | Yes (Duplicates OK) |
| ConsStartYear | Number, Integer | 2 | 2 | No | Yes (Duplicates OK) |
| ConsStartMonth | Number, Integer | 2 | 3 | No | Yes (Duplicates OK) |
| ConsActualFlag | Yes/No | 1 | 4 | No | No |
| ConsNumMonths | Number, Byte | 1 | 5 | No | No |
| ConsAmount | Number, Double | 8 | 6 | No | No |
| ConsDataSourceID | Text | 10 | 7 | No | Yes (Duplicates OK) |
| ConsIflator | Number, Single | 4 | 8 | No | No |
| ConsNote | Memo |  | 9 | No | No |
| ConsDateChanged | Date/Time | 8 | 10 | No | No |
| ConsDisplayNote | Yes/No | 1 | 11 | No | Yes (Duplicates OK) |
| tmpCosts | ProductID | Text | 10 | 1 | No | Yes (Duplicates OK) |
| SupplierID | Text | 10 | 2 | No | Yes (Duplicates OK) |
| dtmEffective | Date/Time | 8 | 3 | No | Yes (Duplicates OK) |
| UnitPrice | Number, Single | 4 | 4 | No | No |
| dtmChanged | Date/Time | 8 | 5 | No | No |
| User | Text | 35 | 6 | No | No |
| Note | Text | 255 | 7 | No | No |
| tmpDataSources | DataSourceID | Text | 10 | 1 | No | Yes (No Duplicates) |
| DataSourceName | Text | 50 | 2 | No | No |
| DataSourceTypeID | Text | 10 | 3 | No | Yes (Duplicates OK) |
| tmpFundingSources | FundingSourceID | Text | 10 | 1 | Yes | Yes (Duplicates OK) |
| FundingSourceName | Text | 50 | 2 | No | No |
| FundingNote | Memo |  | 3 | No | No |
| tmpImp\_Category | MethodID | Text | 50 | 1 | Yes | Yes (Duplicates OK) |
| ParentID | Text | 50 | 2 | No | Yes (Duplicates OK) |
| MethodName | Text | 50 | 3 | No | Yes (No Duplicates) |
| fRollup | Yes/No | 1 | 4 | No | No |
| id | Number, Long Integer | 4 | 5 | No | Yes (No Duplicates) |
| tmpImp\_Country | Code | Text | 2 | 1 | Yes | Yes (Duplicates OK) |
| Country | Text | 50 | 2 | No | No |
| tmpImp\_Price | ProductID | Text | 10 | 1 | Yes | Yes (Duplicates OK) |
| SupplierID | Text | 10 | 2 | Yes | Yes (Duplicates OK) |
| dtmEffective | Date/Time | 8 | 3 | Yes | Yes (Duplicates OK) |
| UnitPrice | Number, Single | 4 | 4 | No | No |
| dtmChanged | Date/Time | 8 | 5 | No | No |
| tmpImp\_Product | ProductID | Text | 10 | 1 | Yes | Yes (Duplicates OK) |
| ProductName | Text | 50 | 2 | No | Yes (No Duplicates) |
| SupplierID | Text | 10 | 3 | No | Yes (Duplicates OK) |
| MethodID | Text | 50 | 4 | No | Yes (Duplicates OK) |
| DefaultCaseSize | Number, Long Integer | 4 | 5 | No | No |
| txtInnovatorDrugName | Text | 50 | 6 | No | No |
| dblLowestUnitQty | Number, Double | 8 | 7 | No | No |
| txtLowestUnitMeasure | Text | 25 | 8 | No | No |
| txtSubstitutionList | Text | 255 | 9 | No | No |
| fPermittedInCountry | Yes/No | 1 | 10 | No | No |
| fAvailabilityStatus | Yes/No | 1 | 11 | No | No |
| txtComments | Text | 50 | 12 | No | No |
| fMapping | Yes/No | 1 | 13 | No | No |
| strMapping | Text | 50 | 14 | No | No |
| strMappingFull | Text | 70 | 15 | No | No |
| fLocked | Yes/No | 1 | 16 | No | No |
| fskipped | Yes/No | 1 | 17 | No | No |
| txtMethodDisplay | Text | 255 | 18 | No | No |
| dtmExported | Date/Time | 8 | 19 | No | No |
| txtSource | Text | 255 | 20 | No | No |
| fSCMSValidity | Yes/No | 1 | 21 | No | No |
| fUserDefined | Yes/No | 1 | 22 | No | No |
| intQuantificationFactor | Number, Long Integer | 4 | 23 | No | No |
| tmpImp\_Subcategory | MethodID | Text | 10 | 1 | Yes | Yes (Duplicates OK) |
| MethodName | Text | 50 | 2 | No | Yes (No Duplicates) |
| ParentID | Text | 50 | 3 | No | Yes (Duplicates OK) |
| fRollup | Yes/No | 1 | 4 | No | No |
| tmpImp\_Supplier | SupplierID | Text | 10 | 1 | Yes | Yes (No Duplicates) |
| SupplierName | Text | 50 | 2 | No | Yes (No Duplicates) |
| tmpImportProducts | strProductID | Text | 20 | 1 | No | Yes (Duplicates OK) |
| strName | Text | 100 | 2 | No | No |
| strDose | Text | 100 | 3 | No | No |
| lngCYP | Number, Double | 8 | 4 | No | No |
| dtmExport | Date/Time | 8 | 5 | No | No |
| fProcessed | Yes/No | 1 | 6 | No | No |
| lngID | Number, Long Integer | 4 | 7 | No | Yes (No Duplicates) |
| strSource | Text | 255 | 8 | No | No |
| strMapping | Text | 50 | 9 | No | No |
| tmpImportProductsSCMS | strProductID | Text | 20 | 1 | No | Yes (Duplicates OK) |
| strName | Text | 100 | 2 | No | No |
| strDose | Text | 100 | 3 | No | No |
| lngCYP | Number, Double | 8 | 4 | No | No |
| dtmExport | Date/Time | 8 | 5 | No | No |
| fProcessed | Yes/No | 1 | 6 | No | No |
| lngID | Number, Long Integer | 4 | 7 | No | Yes (No Duplicates) |
| strSource | Text | 255 | 8 | No | No |
| strMapping | Text | 50 | 9 | No | No |
| tmpImportRecords | Product | Text | 50 | 1 | No | No |
| StartYear | Number, Integer | 2 | 2 | No | No |
| StartMonth | Number, Integer | 2 | 3 | No | No |
| dtmPeriod | Date/Time | 8 | 4 | No | No |
| lngConsumption | Number, Long Integer | 4 | 5 | No | No |
| lngAdjustment | Number, Long Integer | 4 | 6 | No | No |
| dblDataInterval | Number, Double | 8 | 7 | No | No |
| strSource | Text | 255 | 8 | No | No |
| ProductID | Number, Long Integer | 4 | 9 | No | Yes (Duplicates OK) |
| tmpImportRecordsSCMS | Product | Text | 50 | 1 | No | No |
| StartYear | Number, Integer | 2 | 2 | No | No |
| StartMonth | Number, Integer | 2 | 3 | No | No |
| dtmPeriod | Date/Time | 8 | 4 | No | No |
| lngConsumption | Number, Double | 8 | 5 | No | No |
| lngAdjustment | Number, Long Integer | 4 | 6 | No | No |
| dblDataInterval | Number, Double | 8 | 7 | No | No |
| ProductID | Number, Long Integer | 4 | 8 | No | Yes (Duplicates OK) |
| tmpImportRecordsSHIP | ShipmentID | Number, Long Integer | 4 | 1 | No | Yes (Duplicates OK) |
| ProductID | Text | 10 | 2 | No | Yes (Duplicates OK) |
| SupplierID | Text | 10 | 3 | No | Yes (Duplicates OK) |
| ShipAmount | Number, Double | 8 | 4 | No | No |
| ShipPlannedDate | Date/Time | 8 | 5 | No | No |
| ShipReceivedDate | Date/Time | 8 | 6 | No | No |
| ShipStatusCode | Text | 1 | 7 | No | Yes (Duplicates OK) |
| ShipNote | Memo |  | 8 | No | No |
| ShipDateChanged | Date/Time | 8 | 9 | No | No |
| ShipFreightCost | Number, Double | 8 | 10 | No | No |
| ShipValue | Number, Double | 8 | 11 | No | No |
| ShipCaseLot | Number, Long Integer | 4 | 12 | No | No |
| ShipDisplayNote | Yes/No | 1 | 13 | No | No |
| ShipPO | Text | 50 | 14 | No | No |
| strMapping | Text | 50 | 15 | No | No |
| strMappingFull | Text | 70 | 16 | No | No |
| fLocked | Yes/No | 1 | 17 | No | No |
| fDataSourceID | Yes/No | 1 | 18 | No | No |
| fSupplierID | Yes/No | 1 | 19 | No | No |
| fProductID | Yes/No | 1 | 20 | No | No |
| ProductName | Text | 50 | 21 | No | No |
| fMapping | Yes/No | 1 | 22 | No | No |
| dtmExported | Date/Time | 8 | 23 | No | No |
| txtSource | Text | 255 | 24 | No | No |
| ShipDataSourceID | Text | 50 | 25 | No | Yes (Duplicates OK) |
| ShipFundingSourceID | Text | 50 | 26 | No | Yes (Duplicates OK) |
| fFundingSourceID | Yes/No | 1 | 27 | No | No |
| fSplit | Yes/No | 1 | 28 | No | No |
| lngID | Number, Long Integer | 4 | 29 | No | Yes (No Duplicates) |
| tmpInventory | ProductID | Text | 10 | 1 | No | Yes (Duplicates OK) |
| Period | Date/Time | 8 | 2 | No | Yes (Duplicates OK) |
| InvAmount | Number, Double | 8 | 3 | No | No |
| InvTransferFlag | Yes/No | 1 | 4 | No | No |
| InvDataSourceID | Text | 10 | 5 | No | Yes (Duplicates OK) |
| InvNote | Memo |  | 6 | No | No |
| InvDateChanged | Date/Time | 8 | 7 | No | No |
| InvDisplayNote | Yes/No | 1 | 8 | No | No |
| tmpProducts | ProductID | Text | 10 | 1 | No | Yes (No Duplicates) |
| ProductName | Text | 50 | 2 | No | Yes (No Duplicates) |
| ProductMinMonths | Number, Byte | 1 | 3 | No | No |
| ProductMaxMonths | Number, Byte | 1 | 4 | No | No |
| SupplierID | Text | 10 | 5 | No | Yes (Duplicates OK) |
| ProductActiveFlag | Yes/No | 1 | 6 | No | No |
| ProductActiveDate | Date/Time | 8 | 7 | No | No |
| DefaultCaseSize | Number, Long Integer | 4 | 8 | No | No |
| ProductNote | Memo |  | 9 | No | No |
| ProdCMax | Number, Byte | 1 | 10 | No | No |
| ProdCMin | Number, Byte | 1 | 11 | No | No |
| ProdDesStock | Number, Byte | 1 | 12 | No | No |
| MethodID | Text | 10 | 13 | No | Yes (Duplicates OK) |
| tmpProgram | ProgramName | Text | 50 | 1 | No | No |
| DataDirectory | Text | 250 | 2 | No | No |
| Language | Text | 3 | 3 | No | No |
| DefaultLeadTimePlan | Number, Single | 4 | 4 | No | No |
| DefaultLeadTimeOrder | Number, Single | 4 | 5 | No | No |
| DefaultLeadTimeShip | Number, Single | 4 | 6 | No | No |
| DefaultShipCost | Number, Single | 4 | 7 | No | No |
| ProgramContact | Text | 50 | 8 | No | No |
| Telephone | Text | 50 | 9 | No | No |
| Fax | Text | 50 | 10 | No | No |
| Email | Text | 50 | 11 | No | No |
| CountryName | Text | 50 | 12 | No | No |
| IsCurrent | Yes/No | 1 | 13 | No | No |
| Note | Memo |  | 14 | No | No |
| ProgramCode | Text | 50 | 15 | No | Yes (Duplicates OK) |
| IsActive | Yes/No | 1 | 16 | No | No |
| StartSize | Text | 50 | 17 | No | No |
| tmpPTReportTemp | ProductID | Text | 10 | 1 | No | Yes (Duplicates OK) |
| ProductName | Text | 50 | 2 | No | No |
| DesStock | Number, Double | 8 | 3 | No | No |
| tmpShipImp\_Datasources | DataSourceID | Text | 10 | 1 | No | Yes (No Duplicates) |
| DataSourceName | Text | 50 | 2 | No | No |
| DataSourceTypeID | Text | 10 | 3 | No | Yes (Duplicates OK) |
| tmpShipImp\_Fundingsources | FundingSourceID | Text | 10 | 1 | No | Yes (No Duplicates) |
| FundingSourceName | Text | 50 | 2 | No | No |
| tmpShipImp\_Products | ProductID | Text | 10 | 1 | No | Yes (No Duplicates) |
| ProductName | Text | 50 | 2 | No | No |
| ProductMinMonths | Number, Byte | 1 | 3 | No | No |
| ProductMaxMonths | Number, Byte | 1 | 4 | No | No |
| MethodID | Text | 10 | 5 | No | Yes (Duplicates OK) |
| DefaultCaseSize | Number, Long Integer | 4 | 6 | No | No |
| ProductNote | Memo |  | 7 | No | No |
| ProdCMax | Number, Byte | 1 | 8 | No | No |
| ProdCMin | Number, Byte | 1 | 9 | No | No |
| ProdDesStock | Number, Byte | 1 | 10 | No | No |
| tmpShipImp\_Shipments | ShipmentID | Number, Long Integer | 4 | 1 | No | Yes (Duplicates OK) |
| ProductID | Text | 10 | 2 | No | Yes (Duplicates OK) |
| SupplierID | Text | 10 | 3 | No | Yes (Duplicates OK) |
| ShipAmount | Number, Double | 8 | 4 | No | No |
| ShipPlannedDate | Date/Time | 8 | 5 | No | No |
| ShipReceivedDate | Date/Time | 8 | 6 | No | No |
| ShipStatusCode | Text | 1 | 7 | No | Yes (Duplicates OK) |
| ShipNote | Memo |  | 8 | No | No |
| ShipDateChanged | Date/Time | 8 | 9 | No | No |
| ShipFreightCost | Number, Double | 8 | 10 | No | No |
| ShipValue | Number, Double | 8 | 11 | No | No |
| ShipCaseLot | Number, Long Integer | 4 | 12 | No | No |
| ShipDisplayNote | Yes/No | 1 | 13 | No | No |
| ShipPO | Text | 50 | 14 | No | No |
| strMapping | Text | 50 | 15 | No | No |
| strMappingFull | Text | 70 | 16 | No | No |
| fLocked | Yes/No | 1 | 17 | No | No |
| fDataSourceID | Yes/No | 1 | 18 | No | No |
| fSupplierID | Yes/No | 1 | 19 | No | No |
| fProductID | Yes/No | 1 | 20 | No | No |
| ProductName | Text | 50 | 21 | No | No |
| fMapping | Yes/No | 1 | 22 | No | No |
| dtmExported | Date/Time | 8 | 23 | No | No |
| txtSource | Text | 255 | 24 | No | No |
| ShipDataSourceID | Text | 50 | 25 | No | Yes (Duplicates OK) |
| ShipFundingSourceID | Text | 50 | 26 | No | Yes (Duplicates OK) |
| fFundingSourceID | Yes/No | 1 | 27 | No | No |
| lngID | Number, Long Integer | 4 | 28 | No | Yes (No Duplicates) |
| fSplit | Yes/No | 1 | 29 | No | No |
| fProcessed | Yes/No | 1 | 30 | No | No |
| tmpShipImp\_Suppliers | SupplierID | Text | 10 | 1 | No | Yes (No Duplicates) |
| SupplierName | Text | 50 | 2 | No | Yes (No Duplicates) |
| SupplierLeadTimePlan | Number, Single | 4 | 3 | No | No |
| SupplierLeadTimeOrder | Number, Single | 4 | 4 | No | No |
| SupplierLeadTimeShip | Number, Single | 4 | 5 | No | No |
| DefaultSupplier | Yes/No | 1 | 6 | No | Yes (Duplicates OK) |
| tmpShipments | ProductID | Text | 10 | 1 | No | Yes (Duplicates OK) |
| SupplierID | Text | 10 | 2 | No | Yes (Duplicates OK) |
| ShipDataSourceID | Text | 10 | 3 | No | Yes (Duplicates OK) |
| ShipAmount | Number, Double | 8 | 4 | No | No |
| ShipPlannedDate | Date/Time | 8 | 5 | No | No |
| ShipOrderedDate | Date/Time | 8 | 6 | No | No |
| ShipShippedDate | Date/Time | 8 | 7 | No | No |
| ShipReceivedDate | Date/Time | 8 | 8 | No | No |
| ShipStatusCode | Text | 1 | 9 | No | Yes (Duplicates OK) |
| ShipNote | Memo |  | 10 | No | No |
| ShipDateChanged | Date/Time | 8 | 11 | No | No |
| ShipFreightCost | Number, Single | 4 | 12 | No | Yes (Duplicates OK) |
| ShipValue | Number, Double | 8 | 13 | No | Yes (Duplicates OK) |
| ShipCaseLot | Number, Long Integer | 4 | 14 | No | No |
| ShipDisplayNote | Yes/No | 1 | 15 | No | Yes (Duplicates OK) |
| ShipPO | Text | 50 | 16 | No | Yes (Duplicates OK) |
| tmpSortID | CategoryID | Number, Long Integer | 4 | 1 | Yes | Yes (No Duplicates) |
| SortID | Text | 255 | 2 | No | Yes (Duplicates OK) |
| tmpSuppliers | SupplierID | Text | 10 | 1 | No | Yes (No Duplicates) |
| SupplierName | Text | 50 | 2 | No | Yes (No Duplicates) |
| SupplierLeadTimePlan | Number, Single | 4 | 3 | No | No |
| SupplierLeadTimeOrder | Number, Single | 4 | 4 | No | No |
| SupplierLeadTimeShip | Number, Single | 4 | 5 | No | No |
| SupplierNote | Memo |  | 6 | No | No |
| Freight | Number, Single | 4 | 7 | No | No |
| DefaultSupplier | Yes/No | 1 | 8 | No | Yes (Duplicates OK) |
| tmpTypeChanges | MethodID | Text | 10 | 1 | No | Yes (No Duplicates) |
| MethodName | Text | 50 | 2 | No | Yes (No Duplicates) |
| CYPFactor | Number, Single | 4 | 3 | No | No |
| MethodNote | Memo |  | 4 | No | No |
| fRollup | Yes/No | 1 | 5 | No | No |
| ParentID | Text | 255 | 6 | No | Yes (Duplicates OK) |
| CategoryID | Number, Long Integer | 4 | 7 | No | Yes (No Duplicates) |
| tmpXtabStockStatusMatrix | KeyID | Text | 10 | 1 | No | No |
| KeyName | Text | 50 | 2 | No | No |
| Unit | Text | 25 | 3 | No | No |
| StockMin | Number, Byte | 1 | 4 | No | No |
| StockMax | Number, Byte | 1 | 5 | No | No |
| MOS | Number, Double | 8 | 6 | No | No |
| StockDate | Date/Time | 8 | 7 | No | No |
| StockShipSupplier | Text | 10 | 8 | No | No |
| AMCFlag | Number, Long Integer | 4 | 9 | No | No |
| TranslateText | TextID | Text | 6 | 1 | Yes | Yes (Duplicates OK) |
| EnglishText | Text | 255 | 2 | No | No |
| FrenchText | Text | 255 | 3 | No | No |
| SpanishText | Text | 255 | 4 | No | No |
| ArabicText | Text | 255 | 5 | No | No |
| Comment | Text | 50 | 6 | No | No |
| NeedsTranslation | Yes/No | 1 | 7 | No | Yes (Duplicates OK) |
| PortugueseText | Text | 255 | 8 | No | No |
| XtabResult | Column0 | Text | 50 | 1 | No | No |
| Column1 | Text | 50 | 2 | No | No |
| Column2 | Currency | 8 | 3 | No | No |
| Column3 | Currency | 8 | 4 | No | No |
| Column4 | Currency | 8 | 5 | No | No |
| Column5 | Currency | 8 | 6 | No | No |
| Column6 | Currency | 8 | 7 | No | No |
| Column7 | Currency | 8 | 8 | No | No |
| Column8 | Currency | 8 | 9 | No | No |
| Column9 | Currency | 8 | 10 | No | No |
| XtabResultMulti | Column0 | Text | 50 | 1 | No | No |
| Column1 | Text | 50 | 2 | No | No |
| Column2 | Text | 50 | 3 | No | No |
| Column3 | Currency | 8 | 4 | No | No |
| Column4 | Currency | 8 | 5 | No | No |
| Column5 | Currency | 8 | 6 | No | No |
| Column6 | Currency | 8 | 7 | No | No |
| Column7 | Currency | 8 | 8 | No | No |
| Column8 | Currency | 8 | 9 | No | No |
| Column9 | Currency | 8 | 10 | No | No |
| Column10 | Currency | 8 | 11 | No | No |
| XtabResultStockStatus | Column0 | Text | 50 | 1 | No | No |
| Column1 | Text | 50 | 2 | No | No |
| Column2 | Number, Byte | 1 | 3 | No | No |
| Column3 | Number, Byte | 1 | 4 | No | No |
| Column4 | Number, Single | 4 | 5 | No | No |
| Column5 | Number, Single | 4 | 6 | No | No |
| Column6 | Number, Single | 4 | 7 | No | No |
| Column7 | Number, Single | 4 | 8 | No | No |
| Column8 | Number, Single | 4 | 9 | No | No |
| Column9 | Number, Single | 4 | 10 | No | No |
| Column10 | Number, Single | 4 | 11 | No | No |
| Column11 | Number, Single | 4 | 12 | No | No |
| Column12 | Number, Single | 4 | 13 | No | No |
| Column13 | Number, Single | 4 | 14 | No | No |
| Column14 | Number, Single | 4 | 15 | No | No |
| Column15 | Number, Single | 4 | 16 | No | No |
| Column16 | Number, Single | 4 | 17 | No | No |

Program Units

Overview

PipeLine was developed using Microsoft Access based on a split database design. Under this structure the application interface (forms, queries, reports, lookup tables, and code) reside in one database called the front-end and the data database, called the backend.

Source Code CD content

The source code cd contains the application required files, plus installer files, and documentation to assist the developer in understanding PipeLine.

Table - Source Code CD Content

| Directory/File | Purpose |
| --- | --- |
| PL- Module Printout.pdf | Module Printout |
| PL – Procedure List.pdf | Procedure List |
| **/PipeLine** | **Parent directory** |
| **/CODE** | **Code directory** |
| **/ANYMOH** | **Directory for sample database** |
| /globalmoh.MDB | Sample Database |
| **/Graphics** | **Directory for application graphics** |
| /SplashNewT.avi | Splash screen movie |
| /PL40.ico | PipeLine taskbar icon |
| /Pipeline\_ICON-xx.ico | PipeLine Desktop icon |
| **/Summary** | **Directory for PipeLine Summary** |
| /Roboex32.dll | Dll required for PipeLine Summary to run properly |
| /Proc2000.mdb | PipeLine Summary frontend |
| /Proc\_BE.mdb | PipeLine Summary backend |
| /Prog2000.mdb | PipeLine Summary program list |
| /Summary.ico | PipeLine Summary icon |
| /Sumv2.cnt | PipeLine Summary help cnt file |
| /SUMv2.hlp | PipeLine Summary help file |
| **/XML** | **Directory for xml files** |
| /ECatalog\_Live\_Final\_Generic\_20100701.xml | E-Catalog file distributed with application |
| /SCMS Product\_ARV\_TEST.xml | SCMS ARV file distributed with application |
| /Contraceptives.xml | Contraceptives file distributed with application |
| /e-help.cnt | PipeLine (English) help cnt file |
| /e-help.HLP | PipeLine (English) help file |
| /E-PL-help.cnt | PipeLine help cnt file |
| /E-PL-help.hlp | PipeLine help file |
| /Pipeline\_ICON-xx.ico | PipeLine icon |
| /Pipeline2000.MDB | PipeLine frontend file |
| /PLFix1.reg | Registry fix for graphs |
| /PLFix2.reg | Registry fix for graphs |
| /PLFix3.reg | Registry fix for graphs |
| /pmp\_mpty.mdb | Empty PipeLine backend file |
| /posttransform.xslt |  |
| /ProgV4.mdb | PipeLine program list |
| /ReadMe.txt | Readme file for installation issues and known issues |
| /Roboex32.dll | Dll required for PipeLine to run properly |
| **/Help Files** | **Help file directory** |
| **/PipeLine** | **Pipeline directory** |
| **/English** | **English directory** |
| /E-PL-Help.doc | Help word document |
| /E-PL-Help.hpj | Help project |
| **/!SSL!** | **Build directory** |
| **/WinHelp2000** | **WinHelp2000 directory** |
| /e-help.cnt | context file |
| /e-help.hlp | Help file |
| /e-pl-help.cnt | context file |
| / e-pl-help.hlp | Help file |
| / e-pl-help.log | Log file |
| **/French** | **French directory** |
| /E-PL-Help.doc | Help word document |
| /E-PL-Help.hpj | Help project |
| **/Spanish** | **Spanish directory** |
| /E-PL-Help.doc | Help word document |
| /E-PL-Help.hpj | Help project |
| **/Sumv2** | **PL Summary directory** |
| /Sumv2.doc | Help word document |
| / Sumv2.hpj | Help project |
| **/Installer** | **Installer directory** |
| **/Distribution** | **Distribution directory** |
| /accessrt.cab | Access runtime files |
| /Accessrt.msi | Access runtime installer |
| /addendum to user manual for pipeline 5 – English.pdf | Addendum 5.0 |
| /autorun.aru | Autorun file |
| /autorun.ico | Autorun icon |
| /autorun.inf | Autorun ini ile |
| /data1.cab | Added references for installer |
| /install.ini | Installer ini file |
| /msxml6.msi | XML 6 installer |
| /new features and known issues in PipeLine 5.1.pdf | Addendum for 5.1 |
| /pipeline5.cab | Pipeline installer files |
| /pipeline user manual 4 – English.pdf | User Guide for 4.0 |
| /pipeline.msi | Pipeline installer |
| /Pipeline5\_1.exe | Exe that run msi file |
| **/**Readme.txt | Readme text for installer |
| /setup.exe | Setup file for complete installer |
| **/Web** | **Web directory** |
| /About.txt | Text for about statement |
| /PipeLine5\_1Master.zip | Master zip file of installer |
| /password.txt | Text for password warning |
| /pipeline5\_1.exe | Exe file created by winzip |
| /pipeline5\_1.zip | Zip file containing exe and readme file |
| /readme.txt | Readme file for installation |
| **/Schemas** | **Schema directory** |
| **/**MaterialMasterNorm\_070516\_NoCustomType.xsd | Material Master schema |
| **/**SCMgr\_PipeLine\_Export.xsd | SCMgr schema |
| **/**QuantimedForecastOutput.xsd | Quantimed schema |
| **/**PipelineXMLOutputSchema\_070924.xsd | XML export schema |

Program name

The complete Module/Procedure List can be found at:



The Complete Source code:



Forms

Table - List of Forms

| ModuleName | Procedure | Lines | Comments |
| --- | --- | --- | --- |
| fdlgCloneConsumptionSelectDate | (General Declarations) | 1 - 18 | ' Module : fdlgSelectDateRange ' Description: ' Procedures : cmdCancel\_Click ' cmdOK\_Click ' Form\_Open |
| BuildFromYears | 224 - 268 |  |
| cmdCancel\_Click | 87 - 107 | ' Comments : Close the form, which returns the focus ' : to the calling form. ' Parameters : ' Created : 17-Feb-99 Jeff Leiner ' Modified : ' ' -------------------------------------------------------- |
| cmdOK\_Click | 108 - 175 | ' Comments : Check that the data is OK and then set the ' : form's visibility to false to return the focus ' : to the calling form. ' Parameters : ' Created : 17-Feb-99 Jeff Leiner ' Modified : ' ' -------------------------------------------------------- |
| EndYear | 70 - 81 |  |
| Form\_Open | 176 - 211 | ' Comments : Look up the current month and set ' : Date range to a 6 month default. ' Parameters : ' Created : ' Modified : ' ' -------------------------------------------------------- |
| OK | 27 - 37 | '+ ' Property(G): OK ' Comments : ' Parameters : ' Returns : Boolean ' Created : 23 Jul 2009 jleiner ' Modified : '- |
| StartDate | 49 - 60 |  |
| fdlgCloneConsumptionSetRatio | (General Declarations) | 1 - 7 |  |
| CloseMe | 526 - 535 |  |
| cmdCancel\_Click | 29 - 35 |  |
| cmdClone\_Click | 37 - 55 |  |
| cmdPreview\_Click | 58 - 69 |  |
| dblFactor\_AfterUpdate | 73 - 81 |  |
| Form\_GotFocus | 83 - 87 |  |
| Form\_Load | 95 - 116 |  |
| Form\_Open | 520 - 524 |  |
| OK | 16 - 26 | '+ ' Property(G): OK ' Comments : ' Parameters : ' Returns : Boolean ' Created : 17 Jul 2009 jleiner ' Modified : '- |
| ProduceConsumptionRecords | 128 - 253 |  |
| SaveConsumptionRecords | 266 - 518 |  |
| fdlgCountryCode | (General Declarations) | 1 - 2 |  |
| cmdCancel\_Click | 9 - 22 | '+ ' Procedure : cmdCancel\_Click ' Comments : closes form and sends user back to previous screen ' Parameters: - ' Modified : 16 Mar 2007 MAT '- |
| cmdOK\_Click | 45 - 74 | 'place country code into program data table |
| Form\_Open | 30 - 43 |  |
| fdlgCPTGenerator | (General Declarations) | 1 - 21 | ' Module : Form\_fdlgCPTGenerator ' Description: ' Procedures : cboYear\_AfterUpdate() ' cmdCancel\_Click() ' cmdCreateCPTData\_Click() ' Form\_Activate() ' Form\_Load() ' Form\_Open(Cancel As Integer) ' SetComboYears() ' SetText() |
| cboYear\_AfterUpdate | 22 - 46 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| cmdCancel\_Click | 48 - 67 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| cmdCreateCPTData\_Click | 69 - 356 | ' Comments : To Generate the CPT Data files for export to CPT and NewVern. ' : some files will be generated just from the tables data others will ' : be generated due to the products selected in the lstProducts listbox ' Parameters : ' Returns : ' Created : 05/26/98 JSL \* ' Modified : 01 Jul 1998 JSL '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
| Form\_Activate | 358 - 377 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Form\_Load | 379 - 398 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Form\_Open | 400 - 419 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetComboYears | 421 - 474 | ' Comments : To set the range of years to place in the combobox for the CPT ' report. ' Parameters : ' Returns : ' Created : 05/26/98 JSL ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetText | 476 - 496 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| fdlgExportCYP | (General Declarations) | 1 - 14 | ' Module : Form\_fdlgExportCYP ' Description: ' Procedures : SetText() ' cmdCancel\_Click() ' cmdOK\_Click() ' Form\_Open(Cancel As Integer) |
| cmdCancel\_Click | 34 - 51 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| cmdOK\_Click | 53 - 62 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| Form\_Open | 64 - 73 | ' Comments : ' Parameters: Cancel - ' Modified : ' ' -------------------------------------------------- |
| SetText | 15 - 32 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| fdlgForecastMappingSCMS | (General Declarations) | 1 - 18 | '+ ' Module : Form\_fdlgForecastMappingSCMS ' Description: Used for mapping products from SCMS source to PipeLine ' Procedures : ClearData() ' ClearProcessed() ' cmdCancel\_Click() ' cmdOK\_Click() ' Form\_Open(Cancel As Integer) |
| cboSCMSMapping\_AfterUpdate | 44 - 71 | 'this may replace \_beforeUpdate just below |
| cboSCMSMapping\_BeforeUpdate | 73 - 91 | 'if this has a value when the form loads then 'it has to still have a value after update 'in other words, it can't be changed to null if it had value to begin with 'mtracy oct 2006 'Dim fSCMSMapping As Boolean 'fSCMSMapping = False 'default value |
| cboSCMSMapping\_GotFocus | 93 - 100 |  |
| cboSCMSMapping\_LostFocus | 104 - 106 |  |
| ClearUnMapped | 25 - 42 | '+ ' Procedure : ClearUnMapped ' Comments : Removes all unmapped data from the Import tables ' Parameters: - ' Modified : 17 Jul 2006 MAT '- |
| cmdCancel\_Click | 114 - 133 |  |
| cmdOK\_Click | 141 - 219 |  |
| Form\_Open | 226 - 239 | '+ ' Procedure : Form\_Open ' Comments : move to first item in source lookup up and set text ' Parameters: Cancel - ' Modified : 14 Jul 2006 MAT '- |
| ProductNameCut | 289 - 299 |  |
| txtMappingFull\_Click | 244 - 246 |  |
| txtPLMappingFull\_Click | 241 - 243 |  |
| txtPLProduct\_AfterUpdate | 247 - 274 |  |
| txtPLProduct\_GotFocus | 276 - 280 | 'requery all the dropdowns, but don't show items in text boxes |
| txtPLProduct\_LostFocus | 301 - 303 |  |
| fdlgImportForecast | (General Declarations) | 1 - 24 | '+ ' Module : Form\_fdlgImportForecast ' Description: ' Procedures : cmdBrowse\_Click() ' cmdCancel\_Click() ' cmdOK\_Click() ' Form\_Close() ' Form\_Open(Cancel As Integer) |
| cmdBrowse\_Click | 31 - 52 | '+ ' Procedure : cmdBrowse\_Click ' Comments : Opens form for user to select a program file to open ' Parameters: - ' Modified : 27 Jun 2003 LBlanken '- |
| cmdBrowse2\_Click | 60 - 81 |  |
| cmdCancel\_Click | 89 - 105 |  |
| cmdOK\_Click | 113 - 188 |  |
| Form\_Open | 196 - 214 |  |
| fdlgImportMapping | (General Declarations) | 1 - 19 | '+ ' Module : Form\_fdlgImportMapping ' Description: Used for mapping products for external source to PipeLine ' Procedures : cboSelect1\_AfterUpdate() ' ClearData() ' ClearProcessed() ' cmdCancel\_Click() ' cmdOK\_Click() ' Form\_Open(Cancel As Integer) ' ResetValues() |
| cboSelect1\_AfterUpdate | 26 - 38 | '+ ' Procedure : cboSelect1\_AfterUpdate ' Comments : Filters based on source selected ' Parameters: - ' Modified : 21 Aug 2003 LKB '- |
| ClearData | 46 - 69 |  |
| ClearProcessed | 77 - 94 |  |
| cmdCancel\_Click | 102 - 124 |  |
| cmdOK\_Click | 132 - 281 |  |
| Form\_Open | 289 - 313 |  |
| ResetValues | 321 - 336 |  |
| Validate\_Period | 350 - 442 |  |
| fdlgImportMappingSCMS | (General Declarations) | 1 - 16 | '+ ' Module : Form\_fdlgImportMappingSCMS ' Description: Used for mapping products from SCMS source to PipeLine ' Procedures : ClearData() ' ClearProcessed() ' cmdCancel\_Click() ' cmdOK\_Click() ' Form\_Open(Cancel As Integer) |
| cmdCancel\_Click | 23 - 42 | '+ ' Procedure : cmdCancel\_Click ' Comments : resets all values to null (mapping/clear flag) and closes form ' Parameters: - ' Modified : 21 Aug 2003 LKB '- |
| cmdOK\_Click | 50 - 55 |  |
| Form\_Open | 62 - 75 | '+ ' Procedure : Form\_Open ' Comments : move to first item in source lookup up and set text ' Parameters: Cancel - ' Modified : 14 Jul 2006 MAT '- |
| txtMappingFull\_Click | 77 - 79 |  |
| txtProduct\_AfterUpdate | 81 - 104 |  |
| txtProduct\_GotFocus | 106 - 110 | 'requery all the dropdowns, but don't show items in text boxes |
| fdlgImportProducts | (General Declarations) | 1 - 24 | '+ ' Module : Form\_fdlgImportProducts ' Description: ' Procedures : cmdBrowse\_Click() ' cmdCancel\_Click() ' cmdOK\_Click() ' Form\_Close() ' Form\_Open(Cancel As Integer) |
| cmdBrowse\_Click | 31 - 52 | '+ ' Procedure : cmdBrowse\_Click ' Comments : Opens form for user to select a program file to open ' Parameters: - ' Modified : 27 Jun 2003 LBlanken '- |
| cmdCancel\_Click | 60 - 76 |  |
| cmdOK\_Click | 84 - 139 |  |
| Form\_Open | 147 - 164 |  |
| fdlgImportSCM | (General Declarations) | 1 - 24 | '+ ' Module : Form\_fdlgImportSCM ' Description: ' Procedures : cmdBrowse\_Click() ' cmdCancel\_Click() ' cmdOK\_Click() ' Form\_Close() ' Form\_Open(Cancel As Integer) |
| cmdBrowse\_Click | 31 - 52 | '+ ' Procedure : cmdBrowse\_Click ' Comments : Opens form for user to select a program file to open ' Parameters: - ' Modified : 27 Jun 2003 LBlanken '- |
| cmdCancel\_Click | 60 - 76 |  |
| cmdOK\_Click | 84 - 111 |  |
| Form\_Close | 119 - 138 |  |
| Form\_Open | 146 - 159 |  |
| fdlgImportShipmentsPL | (General Declarations) | 1 - 23 | '+ ' Module : Form\_fdlgShipmentsPL ' Description: ' Procedures : cmdBrowse\_Click() ' cmdCancel\_Click() ' cmdOK\_Click() ' Form\_Close() ' Form\_Open(Cancel As Integer) ' Created : 1 Sep 2009 mahmed ' Modified : '- |
| cmdBrowse\_Click | 30 - 51 | '+ ' Procedure : cmdBrowse\_Click ' Comments : Opens form for user to select a program file to open ' Parameters: - ' Modified : '- |
| cmdCancel\_Click | 59 - 75 |  |
| cmdOK\_Click | 83 - 117 |  |
| Form\_Open | 125 - 139 |  |
| fdlgInterpolate | (General Declarations) | 1 - 33 | ' Module : Form\_fdlgCPTGenerator ' Description: ' Procedures : cboYear\_AfterUpdate() ' cmdCancel\_Click() ' cmdCreateCPTData\_Click() ' Form\_Activate() ' Form\_Load() ' Form\_Open(Cancel As Integer) ' SetComboYears() ' SetText() |
| cboMonthStarting\_AfterUpdate | 34 - 42 |  |
| cboYearEnding\_AfterUpdate | 44 - 50 |  |
| cboYearStarting\_AfterUpdate | 52 - 60 |  |
| cmdApply\_Click | 63 - 95 |  |
| cmdCancel\_Click | 101 - 136 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : ' ' -------------------------------------------------------- |
| cmdPreview\_Click | 138 - 172 |  |
| DisplayReport | 618 - 634 |  |
| Form\_Activate | 174 - 193 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : ' ' -------------------------------------------------------- |
| Form\_Load | 195 - 214 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : ' ' -------------------------------------------------------- |
| Form\_Open | 216 - 240 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : ' ' -------------------------------------------------------- |
| lngAMCEnding\_AfterUpdate | 344 - 350 |  |
| lngAMCStarting\_AfterUpdate | 360 - 366 |  |
| SaveData | 380 - 594 |  |
| SetComboYears | 248 - 312 | ' Comments : To set the range of years to place in the combobox for the CPT ' report. ' Parameters : ' Returns : ' Created : ' Modified : ' ' -------------------------------------------------------- |
| SetText | 314 - 334 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : ' ' -------------------------------------------------------- |
| fdlgLanguage | (General Declarations) | 1 - 17 | ' Module : Form\_fdlgLanguage ' Description: ' Procedures : cboSelect\_AfterUpdate() ' cmdAdd\_Click() ' cmdCancel\_Click() ' Form\_Open(Cancel As Integer) ' SetText() |
| cboSelect\_AfterUpdate | 18 - 37 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| cmdAdd\_Click | 39 - 102 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| cmdCancel\_Click | 104 - 130 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Form\_Open | 131 - 152 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetText | 154 - 181 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| fdlgOpenProgram | (General Declarations) | 1 - 23 | '+ ' Module : Form\_fdlgOpenProgram ' Description: Form allows user to select an inactive form to make active ' Procedures : cmdBrowse\_Click() ' cmdCancel\_Click() ' cmdOK\_Click() ' Form\_Open(Cancel As Integer) ' lstSelect\_AfterUpdate() |
| cmdBrowse\_Click | 30 - 53 | '+ ' Procedure : cmdBrowse\_Click ' Comments : Opens form for user to select a program file to open ' Parameters: - ' Modified : 27 Jun 2003 LBlanken '- |
| cmdCancel\_Click | 61 - 76 |  |
| cmdOK\_Click | 84 - 106 |  |
| Form\_Open | 114 - 129 |  |
| lstSelect\_AfterUpdate | 137 - 151 |  |
| fdlgPlanShipmentsByDate | (General Declarations) | 1 - 17 | '+ ' Module : Form\_fdlgPlanShipmentsByDate ' Description: Creates shipments records for products and date entered on form ' Procedures : cmdCancel\_Click() ' cmdOK\_Click() ' GetCaseLot(strProductID As String, strSupplierID As String) |
| cmdCancel\_Click | 24 - 37 | '+ ' Procedure : cmdCancel\_Click ' Comments : Close form w/o doing anything ' Parameters: - ' Modified : 27 Jun 2003 LBlanken '- |
| cmdOK\_Click | 46 - 161 |  |
| Form\_Error | 267 - 300 |  |
| Form\_Open | 302 - 304 |  |
| GetCaseLot | 169 - 238 |  |
| UpdateDate | 239 - 265 |  |
| fdlgProductErrors | (General Declarations) | 1 - 13 | '+ ' Module : Form\_fdlgProductErrors ' Description: Used for making sure imported product names are 50 characters or less ' Procedures : cmdCancel\_Click() ' cmdOK\_Click() ' Form\_Open(Cancel As Integer) |
| cmdCancel\_Click | 20 - 41 | '+ ' Procedure : cmdCancel\_Click ' Comments : resets values, closes form, and sends user back to previous screen ' Parameters: - ' Modified : 9 Feb 2007 MAT '- |
| cmdOK\_Click | 49 - 65 |  |
| Form\_Open | 72 - 90 | '+ ' Procedure : Form\_Open ' Comments : move to first item in source lookup up and set text ' Parameters: Cancel - ' Modified : 14 Jul 2006 MAT '- |
| txtProductNew\_AfterUpdate | 92 - 94 | 'Me.strName = Me.txtProductNew |
| txtProductNew\_BeforeUpdate | 96 - 98 | 'Me.txtProductNew = Left([strName], 50) |
| fdlgProperties | (General Declarations) | 1 - 15 | ' Module : Form\_frmAbout ' Description: ' Procedures : cboLanguage\_AfterUpdate() ' Form\_Open(Cancel As Integer) ' Form\_Timer() ' SetText() |
| cmdClose\_Click | 16 - 18 |  |
| Form\_Open | 20 - 43 | ' Comments : On Open see if an upgrade is needed. If yes ' : give the user the option, if not act as a splash ' : screen. ' Parameters : ' Returns : ' Created : 06/29/98 JSL ' Modified : ' ' -------------------------------------------------------- |
| Form\_Timer | 45 - 69 | ' Comments : if TimerInterval > 0 then count the iterations and ' : do do the correct events ' Parameters : ' Returns : ' Created : 06/29/98 JSL ' Modified : ' ' -------------------------------------------------------- |
| SetText | 71 - 94 | ' Comments : Display the text in the proper language ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| fdlgRecalcAMC | (General Declarations) | 1 - 24 | '+ ' Module : Form\_fdlgStockCount ' Description: Form displays when user click on Enter Count button ' from the stock form. Allows the user to input an ' inventory count and will calculate the adjustment ' based on that value. ' Procedures : cmdCancel\_Click() ' cmdRecalc\_Click() ' cmdSave\_Click() ' Form\_Open(Cancel As Integer) |
| cmdCancel\_Click | 32 - 49 | '+ ' Procedure : cmdCancel\_Click ' Comments : Run when user clicks on cancel button ' Parameters: - ' Modified : 03 Jun 2003 LBlanken ' 16 Jun 2003 LBlanken '- |
| cmdSave\_Click | 59 - 111 |  |
| Form\_Error | 113 - 160 | '+ ' Procedure : Form\_Error ' Comments : Handles validation Errors ' Parameters: DataErr ' Response - ' Modified : 01 Jul 1998 JSL ' 03 Jun 2003 LBlanken ' 16 Jun 2003 LBlanken '- |
| Form\_Open | 169 - 221 |  |
| fdlgSelectDateRange | (General Declarations) | 1 - 12 | ' Module : fdlgSelectDateRange ' Description: ' Procedures : cmdCancel\_Click ' cmdOK\_Click ' Form\_Open |
| BuildFromYears | 132 - 160 |  |
| BuildThroughYears | 171 - 199 | '+ ' Procedure : BuildThroughYears ' Comments : ' Parameters: strType ' frmin ' strIN ' intAdd - ' Returns : - ' Modified : 12 May 2004 LKB ' - |
| cmdCancel\_Click | 13 - 33 | ' Comments : Close the form, which returns the focus ' : to the calling form. ' Parameters : ' Created : 17-Feb-99 Jeff Leiner ' Modified : ' ' -------------------------------------------------------- |
| cmdOK\_Click | 34 - 88 | ' Comments : Check that the data is OK and then set the ' : form's visibility to false to return the focus ' : to the calling form. ' Parameters : ' Created : 17-Feb-99 Jeff Leiner ' Modified : ' ' -------------------------------------------------------- |
| Form\_Open | 89 - 119 | ' Comments : Look up the current month and set ' : Date range to a 6 month default. ' Parameters : ' Created : ' Modified : ' ' -------------------------------------------------------- |
| txtStartYear\_AfterUpdate | 202 - 225 |  |
| fdlgSelectPDFPrinter | (General Declarations) | 1 - 2 |  |
| cmdCancel\_Click | 3 - 8 |  |
| cmdSetPrinter\_Click | 10 - 20 |  |
| Form\_Open | 22 - 27 |  |
| PopulatePrinterList | 29 - 53 |  |
| fdlgSelectRecipient | (General Declarations) | 1 - 15 | ' Module : Form\_fdlgSelectRecipient ' Description: ' Procedures : SetText() ' cmdCancel\_Click() ' cmdOK\_Click() ' Form\_Open(Cancel As Integer) ' lstRecipients\_Click() |
| cmdCancel\_Click | 32 - 41 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| cmdOK\_Click | 43 - 66 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| Form\_Open | 68 - 77 | ' Comments : ' Parameters: Cancel - ' Modified : ' ' -------------------------------------------------- |
| lstRecipients\_Click | 79 - 88 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| SetText | 16 - 30 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| fdlgShipmentMapping | (General Declarations) | 1 - 35 |  |
| checkMappingUsed | 77 - 107 |  |
| chkAllShipments\_AfterUpdate | 36 - 66 |  |
| chkSelect\_AfterUpdate | 122 - 128 |  |
| chkSplit\_AfterUpdate | 138 - 146 |  |
| cmdCancel\_Click | 155 - 172 |  |
| cmdDetail\_Click | 174 - 189 |  |
| cmdOK\_Click | 199 - 207 | 'Save record |
| cmdReport\_Click | 210 - 216 |  |
| cmdShipmentDetail\_Click | 332 - 344 |  |
| Command145\_Click | 348 - 362 |  |
| Form\_Close | 218 - 220 | 'AppCloseEnabled (True) |
| Form\_Open | 229 - 250 |  |
| txtMappingFull\_Click | 252 - 259 |  |
| txtProduct\_AfterUpdate | 262 - 310 | 'End Sub '+ ' Procedure : txtProduct\_Change() ' Comments : reloads all the other dropdowns on the form ' also fills in text box value for said record ' Parameters: - ' Modified : 1 Sep 2009 mahmed '- |
| txtProduct\_GotFocus | 313 - 322 |  |
| fdlgShipmentSearch | (General Declarations) | 1 - 24 | '+ ' Module : Form\_fdlgShipmentSearch ' Description: ' Procedures : cmdSearch\_Click() ' cmdCancel\_Click() ' cmdNext\_Click() ' Form\_Close() ' Form\_Open(Cancel As Integer) |
| cmdCancel\_Click | 31 - 48 | '+ ' Procedure : cmdCancel\_Click ' Comments : Closes the form and returns to shipment module ' Parameters: - ' Modified : 28 July 2009 MTracy '- |
| cmdNext\_Click | 50 - 78 | 'this sub just moves us ahead one rec in the search results 'and then reloads the shipment form with the new product/shipment highlighted |
| cmdSearch\_Click | 86 - 198 |  |
| Form\_Open | 206 - 222 |  |
| fdlgStockCount | (General Declarations) | 1 - 23 | '+ ' Module : Form\_fdlgStockCount ' Description: Form displays when user click on Enter Count button ' from the stock form. Allows the user to input an ' inventory count and will calculate the adjustment ' based on that value. ' Procedures : cmdCancel\_Click() ' cmdRecalc\_Click() ' cmdSave\_Click() ' Form\_Open(Cancel As Integer) |
| cmdCancel\_Click | 31 - 45 | '+ ' Procedure : cmdCancel\_Click ' Comments : Run when user clicks on cancel button ' Parameters: - ' Modified : 03 Jun 2003 LBlanken ' 16 Jun 2003 LBlanken '- |
| cmdRecalc\_Click | 54 - 69 |  |
| cmdSave\_Click | 78 - 122 |  |
| Form\_Error | 124 - 173 | '+ ' Procedure : Form\_Error ' Comments : Handles validation Errors ' Parameters: DataErr ' Response - ' Modified : 01 Jul 1998 JSL ' 03 Jun 2003 LBlanken ' 16 Jun 2003 LBlanken '- |
| Form\_Open | 182 - 213 |  |
| fdlgTextAdmin | (General Declarations) | 1 - 2 |  |
| ArabicStatus\_Change | 22 - 24 |  |
| cmdClose\_Click | 3 - 5 |  |
| EnglishStatus\_Change | 7 - 9 |  |
| Form\_Open | 11 - 13 |  |
| FrenchStatus\_Change | 15 - 17 |  |
| portugueseStatus\_Change | 25 - 27 |  |
| SpanishStatus\_Change | 19 - 21 |  |
| txtDocumentType\_AfterUpdate | 28 - 32 |  |
| txtFormName\_AfterUpdate | 34 - 38 |  |
| fdlgUpdateProducts | (General Declarations) | 1 - 26 | '+ ' Module : Form\_fdlgUpdateProducts ' Description: ' Procedures : cmdBrowse\_Click() ' cmdCancel\_Click() ' cmdOK\_Click() ' Form\_Close() ' Form\_Open(Cancel As Integer) |
| cmdBrowse\_Click | 33 - 54 | '+ ' Procedure : cmdBrowse\_Click ' Comments : Opens form for user to select a program file to open ' Parameters: - ' Modified : 27 Jun 2003 LBlanken '- |
| cmdCancel\_Click | 62 - 78 |  |
| cmdOK\_Click | 86 - 150 |  |
| Form\_Open | 158 - 175 |  |
| frmAbout | (General Declarations) | 1 - 17 | ' Module : Form\_frmAbout ' Description: ' Procedures : cboLanguage\_AfterUpdate() ' Form\_Open(Cancel As Integer) ' Form\_Timer() ' SetText() |
| cboLanguage\_AfterUpdate | 18 - 39 | ' Comments : When the language is modified, update the system ' : variables, and the display. ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| cmdClose\_Click | 42 - 44 |  |
| Form\_Open | 46 - 76 | ' Comments : On Open see if an upgrade is needed. If yes ' : give the user the option, if not act as a splash ' : screen. ' Parameters : ' Returns : ' Created : 06/29/98 JSL ' Modified : ' ' -------------------------------------------------------- |
| Form\_Timer | 78 - 106 | ' Comments : if TimerInterval > 0 then count the iterations and ' : do do the correct events ' Parameters : ' Returns : ' Created : 06/29/98 JSL ' Modified : ' ' -------------------------------------------------------- |
| SetText | 108 - 141 | ' Comments : Display the text in the proper language ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| frmBlank | (General Declarations) | 1 - 2 |  |
| Form\_Open | 3 - 10 |  |
| frmCaseSizes | (General Declarations) | 1 - 27 | ' Module : Form\_frmCaseSizes ' Description: ' Procedures : cboSelect1\_AfterUpdate() ' cboSelect2\_AfterUpdate() ' cmdDelete\_Click() ' cmdNew\_Click() ' cmdSave\_Click() ' Form\_Activate() ' Form\_BeforeUpdate(Cancel As Integer) ' Form\_Delete(Cancel As Integer) ' Form\_Error(DataErr As Integer, Response As Integer) ' Form\_Open(Cancel As Integer) ' lstSelect\_Click() ' SetText() |
| cboSelect1\_AfterUpdate | 28 - 58 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| cboSelect1\_GotFocus | 60 - 82 | ' Comments : ' Parameters : ' Returns : ' Created : 08-Mar-00 Jeff Leiner ' Modified : ' ' -------------------------------------------------------- |
| cboSelect2\_AfterUpdate | 84 - 141 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| cboSelect2\_GotFocus | 143 - 165 | ' Comments : ' Parameters : ' Returns : ' Created : 08-Mar-00 Jeff Leiner ' Modified : ' ' -------------------------------------------------------- |
| cmdDelete\_Click | 167 - 215 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| cmdNew\_Click | 217 - 238 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| cmdSave\_Click | 240 - 284 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| Form\_Activate | 286 - 313 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Form\_BeforeUpdate | 315 - 335 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Form\_Delete | 337 - 362 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Form\_Dirty | 364 - 366 |  |
| Form\_Error | 368 - 411 | ' Comments : ' Parameters : DataErr ' Response ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Form\_Open | 413 - 489 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| lstSelect\_Click | 491 - 519 | ' Comments : ' Parameters : ' Returns : ' Created : 08-Mar-00 Jeff Leiner ' Modified : ' ' -------------------------------------------------------- |
| MovetoClosestCaseSize | 550 - 616 |  |
| SetText | 521 - 541 | ' Comments : To Set each label to the correct text. ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' : 05/18/98 JSL - modified Commodity form to create Case Size ' -------------------------------------------------------- |
| frmCategory | (General Declarations) | 1 - 26 | '+ ' Module : Form\_frmCategory ' Description: Allows users to manage categories throughout system ' Procedures : chkType\_AfterUpdate() ' cmdDelete\_Click() ' cmdNew\_Click() ' cmdSave\_Click() ' Form\_Activate() ' Form\_BeforeUpdate(Cancel As Integer) ' Form\_Dirty(Cancel As Integer) ' Form\_Error(DataErr As Integer, Response As Integer) ' Form\_Open(Cancel As Integer) ' lstSelect\_Click() ' SetText() ' txtName\_AfterUpdate() |
| chkType\_AfterUpdate | 33 - 46 | '+ ' Procedure : chkType\_AfterUpdate ' Comments : Enables the txtCoverage field if true ' Parameters: - ' Modified : 23 Jun 2003 LBlanken '- |
| cmdDelete\_Click | 55 - 112 |  |
| cmdNew\_Click | 121 - 157 |  |
| cmdSave\_Click | 165 - 218 |  |
| Form\_Activate | 226 - 242 |  |
| Form\_AfterUpdate | 243 - 245 |  |
| Form\_BeforeInsert | 246 - 248 |  |
| Form\_BeforeUpdate | 256 - 291 |  |
| Form\_Close | 294 - 303 |  |
| Form\_Dirty | 311 - 324 |  |
| Form\_Error | 334 - 368 |  |
| Form\_Open | 377 - 452 |  |
| lstSelect\_Click | 460 - 496 |  |
| SetText | 505 - 519 |  |
| txtName\_AfterUpdate | 527 - 544 |  |
| txtParent\_AfterUpdate | 546 - 573 |  |
| frmCosts | (General Declarations) | 1 - 29 | ' Module : Form\_frmCosts ' Description: ' Procedures : MarkCurrentPrice() ' cboSelect1\_AfterUpdate() ' cboSelect2\_AfterUpdate() ' cmdDelete\_Click() ' cmdEdit\_Click() ' cmdNew\_Click() ' cmdSave\_Click() ' Form\_Activate() ' Form\_BeforeUpdate(Cancel As Integer) ' Form\_Delete(Cancel As Integer) ' Form\_Error(DataErr As Integer, Response As Integer) ' Form\_Open(Cancel As Integer) ' lstSelect\_Click() ' SetText() |
| cboSelect1\_AfterUpdate | 75 - 104 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| cboSelect1\_GotFocus | 106 - 130 | ' Comments : ' Parameters : ' Returns : ' Created : 08-Mar-00 Jeff Leiner ' Modified : ' ' -------------------------------------------------------- |
| cboSelect2\_AfterUpdate | 132 - 188 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| cboSelect2\_GotFocus | 190 - 214 | ' Comments : ' Parameters : ' Returns : ' Created : 08-Mar-00 Jeff Leiner ' Modified : ' ' -------------------------------------------------------- |
| cmdDelete\_Click | 216 - 266 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| cmdEdit\_Click | 268 - 304 | ' Comments : Edit the selected record in the list box ' Parameters : ' Returns : ' Created : 10-Mar-00 Jeff Leiner ' Modified : ' ' -------------------------------------------------------- |
| cmdNew\_Click | 306 - 329 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| cmdSave\_Click | 331 - 371 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| Form\_Activate | 373 - 401 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Form\_BeforeUpdate | 403 - 422 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Form\_Delete | 424 - 449 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Form\_Dirty | 451 - 453 |  |
| Form\_Error | 455 - 497 | ' Comments : ' Parameters : DataErr ' Response ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Form\_Open | 499 - 579 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| lstSelect\_Click | 581 - 610 | ' Comments : ' Parameters : ' Returns : ' Created : 08-Mar-00 Jeff Leiner ' Modified : ' ' -------------------------------------------------------- |
| MarkCurrentPrice | 30 - 73 | ' Comments : For the supplier / product Selected, Select the currently ' : Active Price in the listbox ' Parameters : ' Returns : ' Created : 10-Mar-00 Jeff Leiner ' Modified : ' ' -------------------------------------------------------- |
| SetText | 612 - 632 | ' Comments : To Set each label to the correct text. ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' : 05/18/98 JSL - modified Commodity form to create Case Size ' -------------------------------------------------------- |
| frmDataSources | (General Declarations) | 1 - 32 | '+ ' Module : Form\_frmDataSources ' Description: Form allowing user to modify data sources for various types ' Procedures : cboSelect1\_AfterUpdate() ' cboSelect1\_GotFocus() ' cmdDelete\_Click() ' cmdNew\_Click() ' cmdSave\_Click() ' Form\_Activate() ' Form\_BeforeInsert(Cancel As Integer) ' Form\_BeforeUpdate(Cancel As Integer) ' Form\_Dirty(Cancel As Integer) ' Form\_Error(DataErr As Integer, Response As Integer) ' Form\_Open(Cancel As Integer) ' lstSelect\_Click() ' SetText() ' txtCode\_BeforeUpdate(Cancel As Integer) ' txtName\_AfterUpdate() |
| cboSelect1\_AfterUpdate | 40 - 64 | '+ ' Procedure : cboSelect1\_AfterUpdate ' Comments : Runs after the user selects a data source type ' Parameters: - ' Modified : 03 Jun 2003 LBlanken ' 16 Jun 2003 LBlanken '- |
| cboSelect1\_GotFocus | 74 - 93 |  |
| cmdDelete\_Click | 103 - 157 |  |
| cmdNew\_Click | 167 - 198 |  |
| cmdSave\_Click | 207 - 258 |  |
| Form\_Activate | 267 - 286 |  |
| Form\_BeforeInsert | 295 - 310 |  |
| Form\_BeforeUpdate | 319 - 355 |  |
| Form\_Dirty | 364 - 379 |  |
| Form\_Error | 389 - 423 |  |
| Form\_Open | 433 - 473 |  |
| lstSelect\_Click | 482 - 510 |  |
| SetText | 520 - 534 |  |
| txtCode\_BeforeUpdate | 545 - 604 |  |
| txtName\_AfterUpdate | 613 - 629 |  |
| frmFundingSources | (General Declarations) | 1 - 33 | '+ ' Module : Form\_frmFundingSources ' Description: Form allowing user to modify Funding Sources for various types ' Procedures : ' ' cmdDelete\_Click() ' cmdNew\_Click() ' cmdSave\_Click() ' Form\_Activate() ' Form\_BeforeInsert(Cancel As Integer) ' Form\_BeforeUpdate(Cancel As Integer) ' Form\_Dirty(Cancel As Integer) ' Form\_Error(DataErr As Integer, Response As Integer) ' Form\_Open(Cancel As Integer) ' lstSelect\_Click() ' SetText() ' txtCode\_BeforeUpdate(Cancel As Integer) ' txtName\_AfterUpdate() |
| cmdDelete\_Click | 42 - 96 | '+ ' Procedure : cmdDelete\_Click ' Comments : Runs when the user clicks on the delete button ' Parameters: - ' Modified : 01 Jul 1998 JSL ' 03 Jun 2003 LBlanken ' 16 Jun 2003 LBlanken '- |
| cmdNew\_Click | 106 - 137 |  |
| cmdSave\_Click | 145 - 196 |  |
| Form\_Activate | 205 - 224 |  |
| Form\_BeforeUpdate | 233 - 269 |  |
| Form\_Dirty | 278 - 293 |  |
| Form\_Error | 303 - 337 |  |
| Form\_Open | 347 - 379 |  |
| lstSelect\_Click | 388 - 416 |  |
| SetText | 426 - 440 |  |
| txtCode\_BeforeUpdate | 451 - 501 |  |
| txtName\_AfterUpdate | 510 - 526 |  |
| frmGraphConsumption | (General Declarations) | 1 - 33 | '+ ' Module : Form\_frmGraphConsumption ' Description: ' Procedures : cboDisplay\_AfterUpdate() ' cboFrom\_AfterUpdate() ' cboGrouping\_AfterUpdate() ' cboThrough\_AfterUpdate() ' cmdPreview\_Click() ' cmdPrint\_Click() ' DisplayGraph() ' DisplayReport(intPrint As Integer) ' Form\_Activate() ' Form\_Open(Cancel As Integer) ' GetReportSQL() ' ini\_cboProduct() ' ini\_cboYears() ' lstDisplay1\_AfterUpdate() ' optDisplay\_AfterUpdate() ' SetText() ' update\_Graph() ' UpdateControls(lngColumn As Long) |
| cboDisplay\_AfterUpdate | 41 - 59 | '+ ' Procedure : cboDisplay\_AfterUpdate ' Comments : ' Parameters: - ' Modified : 11 Jul 2003 LBlanken ' 22 Dec 2003 LBlanken '- |
| cboFrom\_AfterUpdate | 70 - 97 |  |
| cboGrouping\_AfterUpdate | 106 - 131 |  |
| cboPeriod\_AfterUpdate | 133 - 163 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 15 Jul 2009 mahmed ' ' -------------------------------------------------------- |
| cboPeriod\_GotFocus | 166 - 168 |  |
| cboPeriod\_MouseDown | 170 - 172 | 'Call SetDropDownWidth([cboPeriod]) |
| cboThrough\_AfterUpdate | 181 - 199 |  |
| cmdPDF\_Click | 201 - 230 | '+ ' Procedure : cmdPDF\_Click ' Comments : ' Parameters: - ' Modified : 01 Jul 1998 JSL ' Modified : 11 Jul 2003 LBlanken '- |
| cmdPreview\_Click | 239 - 264 |  |
| cmdPrint\_Click | 273 - 297 |  |
| cmdShowHide\_Click | 517 - 522 |  |
| DisplayGraph | 306 - 422 |  |
| DisplayReport | 431 - 514 |  |
| Form\_Activate | 531 - 543 |  |
| Form\_Open | 553 - 646 |  |
| GetReportSQL | 655 - 773 | ' Comments : Update the Rowsource for the Graph object ' Parameters : ' Returns : ' Created : 08/17/98 JSL ' Modified : ' ' -------------------------------------------------------- |
| ini\_cboProduct | 782 - 860 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- ' Set up lstDisplay1 with ProductID of first record in Consumption |
| ini\_cboYears | 869 - 914 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- ' Set up cboStart/End Year boxes from MonthlyStock |
| lstDisplay1\_AfterUpdate | 922 - 1005 |  |
| lstDisplay1\_KeyDown | 1007 - 1009 |  |
| lstDisplay1\_KeyUp | 1011 - 1013 |  |
| optBU\_AfterUpdate | 1021 - 1034 |  |
| optDisplay\_AfterUpdate | 1043 - 1133 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| SetText | 1141 - 1167 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| ShowFirstSelected | 1285 - 1339 |  |
| update\_Graph | 1175 - 1239 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| UpdateControls | 1248 - 1284 |  |
| frmGraphCYP | (General Declarations) | 1 - 29 | '+ ' Module : Form\_frmGraphCYP ' Description: ' Procedures : getSelectedProducts() ' cboFrom\_AfterUpdate() ' cboGrouping\_AfterUpdate() ' cboThrough\_AfterUpdate() ' cmdExport\_Click() ' cmdPreview\_Click() ' cmdPrint\_Click() ' DisplayGraph() ' DisplayReport(intPrint As Integer) ' Form\_Activate() ' Form\_Open(Cancel As Integer) ' ini\_cboProduct() ' ini\_cboYears() ' lstDisplay\_AfterUpdate() ' optDisplay\_AfterUpdate() ' SetText() ' update\_Graph() |
| cboFrom\_AfterUpdate | 98 - 131 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| cboGrouping\_AfterUpdate | 139 - 170 | ' Comments : ' Parameters : ' Returns : ' Created : 08/31/98 JSL ' Modified : ' ' -------------------------------------------------------- |
| cboPeriod\_AfterUpdate | 172 - 203 |  |
| cboPeriod\_GotFocus | 206 - 208 |  |
| cboThrough\_AfterUpdate | 216 - 240 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| CmdExport\_Click | 248 - 318 | ' Comments : Export the selected data to an Excel Spreadsheet for use ' : in an FPPMES Database. ' Parameters : ' Returns : ' Created : 10-Mar-00 Jeff Leiner ' Modified : ' ' -------------------------------------------------------- |
| cmdPDF\_Click | 320 - 349 | '+ ' Procedure : cmdPDF\_Click ' Comments : ' Parameters: - ' Modified : 11 Jul 2003 LBlanken '- |
| cmdPreview\_Click | 357 - 387 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| cmdPrint\_Click | 395 - 426 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| cmdShowHide\_Click | 613 - 618 |  |
| DisplayGraph | 434 - 548 | ' Comments : Update the Rowsource for the Graph object ' Parameters : ' Returns : ' Created : 08/17/98 JSL ' Modified : ' ' -------------------------------------------------------- |
| DisplayReport | 556 - 610 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Form\_Activate | 626 - 646 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Form\_Error | 656 - 686 |  |
| Form\_Open | 694 - 776 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- ' Set up cboProductID with ProductID of first record in Inventory |
| getSelectedProducts | 37 - 90 | '+ ' Procedure : getSelectedProducts ' Comments : ' Parameters: - ' Returns : String - ' Modified : 11 Jul 2003 LBlanken ' - |
| ini\_cboProduct | 785 - 851 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- ' Set up lstDisplay with ProductID of first record in Consumption |
| ini\_cboYears | 860 - 905 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- ' Set up cboStart/End Year boxes from MonthlyStock |
| lstDisplay\_AfterUpdate | 913 - 956 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| lstDisplay\_KeyDown | 958 - 960 |  |
| lstDisplay\_KeyUp | 962 - 964 |  |
| optBU\_AfterUpdate | 972 - 985 |  |
| optDisplay\_AfterUpdate | 993 - 1071 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| SetText | 1079 - 1107 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| ShowFirstSelected | 1197 - 1251 |  |
| update\_Graph | 1115 - 1195 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| frmGraphStockStatus | (General Declarations) | 1 - 37 | '+ ' Module : Form\_frmGraphStockStatus ' Description: ' Procedures : SetListboxSelected(lstBox As ListBox, fSelected As Boolean) ' cboDisplay\_AfterUpdate() ' cboFrom\_AfterUpdate() ' cboGrouping\_AfterUpdate() ' cboThrough\_AfterUpdate() ' chkAllSupplier\_AfterUpdate() ' cmdPreview\_Click() ' cmdPrint\_Click() ' DisplayGraph() ' DisplayReport(intPrint As Integer) ' Form\_Activate() ' Form\_Open(Cancel As Integer) ' GetReportSQL() ' ini\_cboProduct() ' ini\_cboYears() ' lstDisplay1\_AfterUpdate() ' lstDisplay2\_AfterUpdate() ' optDisplay\_AfterUpdate() ' SetText() ' update\_Graph() ' UpdateControls(lngColumn As Long) |
| cboDisplay\_AfterUpdate | 80 - 98 |  |
| cboFrom\_AfterUpdate | 106 - 142 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' Modified : 14 July 2009 mahmed ' set and display user selected date range ' -------------------------------------------------------- |
| cboGrouping\_AfterUpdate | 150 - 181 | ' Comments : ' Parameters : ' Returns : ' Created : 08/31/98 JSL ' Modified : ' ' -------------------------------------------------------- |
| cboPeriod\_AfterUpdate | 183 - 213 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 15 Jul 2009 mahmed ' ' -------------------------------------------------------- |
| cboPeriod\_GotFocus | 216 - 218 |  |
| cboThrough\_AfterUpdate | 226 - 252 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| chkAllSupplier\_AfterUpdate | 260 - 288 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| cmdPDF\_Click | 290 - 321 | '+ ' Procedure : cmdPDF\_Click ' Comments : ' Parameters: - ' Modified : 11 Jul 2003 LBlanken '- |
| cmdPreview\_Click | 329 - 362 | ' Comments : Open the Status Graph Report (in Preview mode) ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' -------------------------------------------------------- |
| cmdPrint\_Click | 370 - 402 | ' Comments : Open the Status Graph Report (in Preview mode) ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| cmdShowHide\_Click | 602 - 607 |  |
| DisplayGraph | 410 - 518 | ' Comments : Update the Rowsource for the Graph object ' Parameters : ' Returns : ' Created : 08/17/98 JSL ' Modified : ' ' -------------------------------------------------------- |
| DisplayReport | 526 - 599 | ' Comments : Open the Status Graph Report (in Preview mode) ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Form\_Activate | 615 - 633 | ' Comments : Reset the text. ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Form\_Error | 643 - 673 |  |
| Form\_Open | 683 - 783 |  |
| GetReportSQL | 792 - 890 | ' Comments : Update the Rowsource for the Graph object ' Parameters : ' Returns : ' Created : 08/17/98 JSL ' Modified : ' ' -------------------------------------------------------- |
| ini\_cboProduct | 899 - 980 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- ' Set up lstDisplay1 with ProductID of first record in Consumption |
| ini\_cboYears | 989 - 1040 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- ' Set up cboStart/End Year boxes from MonthlyStock |
| lstDisplay1\_AfterUpdate | 1048 - 1136 |  |
| lstDisplay1\_KeyDown | 1138 - 1140 |  |
| lstDisplay1\_KeyUp | 1142 - 1144 |  |
| lstDisplay2\_AfterUpdate | 1152 - 1176 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| lstDisplay2\_KeyDown | 1182 - 1184 |  |
| lstDisplay2\_KeyUp | 1186 - 1188 |  |
| optBU\_AfterUpdate | 1178 - 1180 |  |
| optDisplay\_AfterUpdate | 1197 - 1282 |  |
| SetListboxSelected | 45 - 71 | '+ ' Procedure : SetListboxSelected ' Comments : ' Parameters: lstBox ' fSelected - ' Modified : 11 Jul 2003 LBlanken '- |
| SetText | 1290 - 1317 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| ShowFirstSelected | 1427 - 1481 |  |
| update\_Graph | 1325 - 1379 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| UpdateControls | 1388 - 1425 |  |
| frmInvCount | (General Declarations) | 1 - 30 | ' Module : Form\_frmInvCount ' Description: ' Procedures : FindClosestRecord() ' cboSelect2\_AfterUpdate() ' cmdDelete\_Click() ' cmdNew\_Click() ' cmdSave\_Click() ' Form\_Activate() ' Form\_BeforeInsert(Cancel As Integer) ' Form\_BeforeUpdate(Cancel As Integer) ' Form\_Error(DataErr As Integer, Response As Integer) ' Form\_Open(Cancel As Integer) ' lstSelect\_Click() ' SetText() ' txtDate\_AfterUpdate() ' txtDate\_BeforeUpdate(Cancel As Integer) |
| cboSelect2\_AfterUpdate | 90 - 102 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| cboSelect2\_GotFocus | 104 - 127 | ' Comments : ' Parameters : ' Returns : ' Created : 08-Mar-00 Jeff Leiner ' Modified : ' ' -------------------------------------------------------- |
| cmdDelete\_Click | 129 - 183 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| cmdNew\_Click | 185 - 211 | ' Comments : Open the Detail Form to A new record ' Parameters : ' Returns : ' Created : 07-Feb-00 Jeff Leiner ' Modified : ' ' -------------------------------------------------------- |
| cmdSave\_Click | 213 - 250 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| FindClosestRecord | 31 - 88 | ' Comments : ' Parameters : ' Returns : ' Created : 04-Feb-00 Jeff Leiner ' Modified : ' ' -------------------------------------------------------- |
| Form\_Activate | 252 - 261 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| Form\_BeforeInsert | 263 - 278 | ' Comments : ' Parameters: Cancel - ' Modified : ' ' -------------------------------------------------- |
| Form\_BeforeUpdate | 280 - 329 | ' Comments : ' Parameters: Cancel - ' Modified : ' ' -------------------------------------------------- |
| Form\_Dirty | 331 - 333 |  |
| Form\_Error | 335 - 387 | ' Comments : Handles validation Errors ' Parameters : DataErr ' Response ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Form\_Open | 389 - 519 | ' Comments : ' Parameters : cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- ' Set up cboSelect2 with ProductID of first record in Consumption |
| lstSelect\_Click | 521 - 536 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| SetText | 538 - 557 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| txtDate\_AfterUpdate | 559 - 613 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| txtDate\_BeforeUpdate | 615 - 642 | ' Comments : See if a record exists for the txtDate already ' Parameters : ' Returns : ' Created : 07/13/98 JSL ' Modified : ' ' -------------------------------------------------------- |
| frmMain | (General Declarations) | 1 - 22 | '+ ' Module : Form\_frmMain ' Description: ' Procedures : cboLanguage\_AfterUpdate() ' Form\_Open(Cancel As Integer) ' SetText() ' Treeview1\_GotFocus() ' Treeview1\_LostFocus() ' Treeview1\_NodeClick(ByVal Node As Object) |
| cboLanguage\_AfterUpdate | 30 - 48 | '+ ' Procedure : cboLanguage\_AfterUpdate ' Comments : Updates language based on value ' Parameters: - ' Modified : 01 Jul 1998 JSL ' 16 Jun 2003 LBlanken '- |
| Form\_Activate | 60 - 90 |  |
| Form\_Close | 92 - 95 |  |
| Form\_Open | 104 - 153 |  |
| RequeryConsumption | 344 - 346 |  |
| SetText | 161 - 179 |  |
| Treeview1\_GotFocus | 187 - 202 |  |
| Treeview1\_LostFocus | 210 - 224 |  |
| Treeview1\_NodeClick | 233 - 339 |  |
| UpdateLanguage | 340 - 342 |  |
| frmMain\_temp | (General Declarations) | 1 - 22 | '+ ' Module : Form\_frmMain ' Description: ' Procedures : cboLanguage\_AfterUpdate() ' Form\_Open(Cancel As Integer) ' SetText() ' Treeview1\_GotFocus() ' Treeview1\_LostFocus() ' Treeview1\_NodeClick(ByVal Node As Object) |
| cboLanguage\_AfterUpdate | 30 - 48 | '+ ' Procedure : cboLanguage\_AfterUpdate ' Comments : Updates language based on value ' Parameters: - ' Modified : 01 Jul 1998 JSL ' 16 Jun 2003 LBlanken '- |
| Form\_Close | 53 - 56 |  |
| Form\_Open | 65 - 117 |  |
| SetText | 125 - 143 |  |
| Treeview1\_GotFocus | 151 - 166 |  |
| Treeview1\_LostFocus | 174 - 188 |  |
| Treeview1\_NodeClick | 197 - 298 |  |
| UpdateLanguage | 299 - 301 |  |
| frmProducts | (General Declarations) | 1 - 26 | ' Module : Form\_frmProducts ' Description: Allows users to maintain the list of products used ' throughout the system. ' Procedures : CheckMinMaxs() ' cmdCategories\_Click() ' cmdDelete\_Click() ' cmdNew\_Click() ' cmdSave\_Click() ' Form\_Activate() ' Form\_AfterUpdate() ' Form\_BeforeInsert(Cancel As Integer) ' Form\_BeforeUpdate(Cancel As Integer) ' Form\_Dirty(Cancel As Integer) ' Form\_Open(Cancel As Integer) ' Form\_Unload(Cancel As Integer) ' lstSelect\_Click() ' SetText() ' txtName\_AfterUpdate() ' Modified : 06/03/03 lkb ' 05/22/03 lkb Cleaned with Total Visual CodeTools ' -------------------------------------------------- |
| checkDEOYS | 27 - 50 | ' Comments : Compare the Program Min and Max to the DEOYS ' : and make sure DEOYS is within the two. ' Parameters : - ' Returns : Boolean ' Created : 13-Jun-06 mtracy ' Modified : ' -------------------------------------------------------- |
| CheckMinMaxs | 52 - 144 | ' Comments : Compare the Product Mins and Maxs and the Central level ' : Mins and maxs and make sure they all fit in the proper ' : intervals. ' Parameters : - ' Returns : Boolean ' Created : 25-Feb-00 Jeff Leiner ' Modified : ' -------------------------------------------------------- |
| cmdCategories\_Click | 146 - 153 | ' Comments : opens the assign category form ' Parameters: - ' Modified : 06/03/03 lkb ' -------------------------------------------------- 'open popup form |
| cmdDelete\_Click | 155 - 224 | ' Comments : deletes selected item in listbox ' Parameters : - ' Modified : 07/01/98 JSL ' 06/03/03 lkb ' -------------------------------------------------------- |
| cmdImport\_Click | 226 - 229 |  |
| cmdNew\_Click | 231 - 266 | ' Comments : Open the Detail Form to A new record ' Parameters : - ' Returns : - ' Created : 07-Feb-00 Jeff Leiner ' Modified : 06/03/03 lkb ' -------------------------------------------------------- |
| cmdSave\_Click | 268 - 327 | ' Comments : saves the recod and updates listbox/form ' Parameters: - ' Modified : ' -------------------------------------------------- |
| Form\_Activate | 329 - 350 | ' Comments : sets value of list box ' Parameters: - ' Modified : 06/03/03 lkb ' -------------------------------------------------- |
| Form\_AfterUpdate | 352 - 363 | ' Comments : sets global product id ' Parameters: - ' Modified : 06/03/03 lkb ' -------------------------------------------------- |
| Form\_BeforeInsert | 365 - 391 | ' Comments : Find the Default Supplier ' Parameters : - ' Returns : - ' Created : 07/13/98 JSL ' Modified : ' -------------------------------------------------------- |
| Form\_BeforeUpdate | 393 - 447 | ' Comments : validates data ' Parameters: Cancel - exit process ' Modified : 06/03/03 lkb ' -------------------------------------------------- |
| Form\_Dirty | 449 - 455 | ' Comments : sets global dirty flag ' Parameters: Cancel - exit process ' Modified : 06/03/03 lkb ' -------------------------------------------------- |
| Form\_Error | 465 - 510 |  |
| Form\_Open | 512 - 575 | ' Comments : sets rowsource for listbox, set value of listbox ' to global product ID (if global is null, sets it) ' Parameters : cancel - exit process ' Returns : - ' Modified : 07/01/98 JSL ' 06/03/03 lkb ' -------------------------------------------------------- |
| Form\_Unload | 577 - 619 | ' Comments : if the current product isn't active, Check to see ' : if there is at least one active product ' Parameters : cancel - exit process ' Returns : - ' Created : 18-Feb-00 Jeff Leiner ' Modified : ' -------------------------------------------------------- |
| lstSelect\_Click | 621 - 650 | ' Comments : Prompt user to save form if dirty, else requery form ' Parameters: - ' Modified : 06/03/03 lkb ' -------------------------------------------------- |
| SetText | 652 - 674 | ' Comments : sets the captions, statusbar, styles for controls on form ' Parameters : - ' Modified : 07/01/98 JSL ' 06/03/03 lkb ' -------------------------------------------------------- |
| txtDesCMax\_AfterUpdate | 676 - 680 | 'this is causing too many validations and making it difficult to add data 'therefore I'm moving it to run on save. (LKB 4/4/2007) 'CheckMinMaxs Me!txtDesCMax, False |
| txtDesCMin\_AfterUpdate | 682 - 686 | 'this is causing too many validations and making it difficult to add data 'therefore I'm moving it to run on save. (LKB 4/4/2007) 'CheckMinMaxs Me!txtDesCMin, False |
| txtDesMax\_AfterUpdate | 688 - 693 |  |
| txtDesMin\_AfterUpdate | 695 - 700 |  |
| txtDesStock\_AfterUpdate | 702 - 708 |  |
| txtName\_AfterUpdate | 710 - 721 | ' Comments : generate codes after name is entered ' Parameters: - ' Modified : 06/03/03 lkb ' -------------------------------------------------- |
| UpdateControls | 723 - 758 |  |
| frmProgram | (General Declarations) | 1 - 53 | '+ ' Module : Form\_frmProgram ' Description: Manages program form ' Procedures : IsImportVisible() ' SetFormMode() ' cboDisplay\_AfterUpdate() ' cboDisplay\_GotFocus() ' cmdAdd\_Click() ' cmdBackup\_Click() ' cmdCPTExport\_Click() ' cmdDelete\_Click() ' cmdImport\_Click() ' cmdImport\_GotFocus() ' cmdImportShipmentData\_Click() ' cmdImportShipmentData\_GotFocus() ' cmdRestore\_Click() ' DirtyContinue() ' edit\_error() ' Form\_AfterInsert() ' Form\_AfterUpdate() ' Form\_BeforeUpdate(Cancel As Integer) ' Form\_Current() ' Form\_Delete(Cancel As Integer) ' Form\_Dirty(Cancel As Integer) ' Form\_Error(DataErr As Integer, Response As Integer) ' Form\_Open(Cancel As Integer) ' Form\_Unload(Cancel As Integer) ' LockFields(intLock As Integer) ' MakeActive(fActive As Boolean, strCode As String) ' MakeCurrent() ' OpenDialog(strFrmName As String) ' ResetStatus() ' SetText() ' txtBlocker\_Enter() ' UpdateBackEnd() |
| cboCountryCode\_AfterUpdate | 54 - 67 |  |
| cboCountryCode\_NotInList | 69 - 74 | 'mtracy 4/20/07 for bugzilla 14546 |
| cboDisplay\_GotFocus | 82 - 97 |  |
| cboLanguage\_AfterUpdate | 100 - 102 |  |
| cboLanguage\_GotFocus | 104 - 106 |  |
| cboLanguage\_NotInList | 108 - 114 | 'LBLANKEN 4/20/07 for bugzilla 14546 |
| cmdAdd\_Click | 124 - 207 |  |
| cmdBackup\_Click | 217 - 262 |  |
| edit\_error | 272 - 339 |  |
| Form\_AfterInsert | 348 - 374 |  |
| Form\_AfterUpdate | 383 - 416 |  |
| Form\_BeforeUpdate | 425 - 532 |  |
| Form\_Close | 535 - 545 |  |
| Form\_Current | 554 - 576 |  |
| Form\_Dirty | 584 - 598 |  |
| Form\_Error | 608 - 666 |  |
| Form\_Open | 675 - 761 |  |
| Form\_Unload | 770 - 793 |  |
| LockFields | 802 - 842 |  |
| OpenDialog | 851 - 863 |  |
| ResetStatus | 872 - 884 |  |
| txtBlocker\_Enter | 893 - 906 |  |
| txtCode\_AfterUpdate | 908 - 912 |  |
| frmRptAnnualCosts | (General Declarations) | 1 - 23 | '+ ' Module : Form\_frmRptAnnualCosts ' Description: Code for Annual Cost Report parameter form ' Procedures : cboFrom\_AfterUpdate() ' cboThrough\_AfterUpdate() ' cmdPreview\_Click() ' cmdPrint\_Click() ' FillCombos(ctlCombo As Control, intMedian As Integer, intStart As Integer, intEnd As Integer) ' FillFilterCombos(strType As String) ' Form\_Load() ' Form\_Open(Cancel As Integer) ' MakeDataSet() ' SetText() |
| cboDisplay\_AfterUpdate | 24 - 69 |  |
| cboFrom\_AfterUpdate | 78 - 92 |  |
| cboPeriod\_AfterUpdate | 95 - 110 |  |
| cboPeriod\_GotFocus | 113 - 115 |  |
| cboThrough\_AfterUpdate | 124 - 140 |  |
| cmdPDF\_Click | 142 - 174 | '+ ' Procedure : cmdPreview\_Click ' Comments : diplay report in preview mode ' Parameters: - ' Modified : 01 Jul 1998 JSL ' 11 Jul 2003 LBlanken '- |
| cmdPreview\_Click | 183 - 208 |  |
| cmdPrint\_Click | 217 - 236 |  |
| FillCombos | 249 - 296 |  |
| FillFilterCombos | 306 - 421 |  |
| Form\_Error | 431 - 461 |  |
| Form\_Load | 470 - 498 |  |
| Form\_Open | 507 - 543 |  |
| lstDisplay1\_AfterUpdate | 672 - 701 |  |
| lstDisplay2\_AfterUpdate | 703 - 726 |  |
| MakeDataSet | 553 - 642 |  |
| SetText | 651 - 670 |  |
| frmRptPipelineAction | (General Declarations) | 1 - 19 | '+ ' Module : Form\_frmRptPipelineAction ' Description: Code for Pipeline Action Report parameter form ' Procedures : cmdPreview\_Click() ' cmdPrint\_Click() ' cmdShowHide\_Click() ' DisplayReport(intView As Integer) ' Form\_Open(Cancel As Integer) ' SetText() |
| cboDisplay\_AfterUpdate | 20 - 42 |  |
| cboFrom\_AfterUpdate | 50 - 75 |  |
| cboPeriod\_AfterUpdate | 76 - 98 |  |
| cboPeriod\_GotFocus | 101 - 103 |  |
| cboThrough\_AfterUpdate | 111 - 127 |  |
| cmdPDF\_Click | 129 - 148 | '+ ' Procedure : cmdPDF\_Click ' Comments : display report in preview mode ' Parameters: - ' Modified : 01 Jul 1998 JSL ' 11 Jul 2003 LBlanken '- |
| cmdPreview\_Click | 157 - 170 |  |
| cmdPrint\_Click | 179 - 192 |  |
| cmdShowHide\_Click | 200 - 229 |  |
| DisplayReport | 238 - 314 |  |
| Form\_Error | 324 - 354 |  |
| Form\_Open | 362 - 401 |  |
| SetText | 410 - 428 |  |
| VerifyDateRange | 438 - 477 |  |
| frmRptPipelineProblem | (General Declarations) | 1 - 20 | '+ ' Module : Form\_frmRptPipelineProblem ' Description: Code for Pipeline Problem Report parameter form ' Procedures : cmdPreview\_Click() ' cmdPrint\_Click() ' DisplayReport(intView As Integer) ' Form\_Open(Cancel As Integer) ' SetText() |
| cboDisplay\_AfterUpdate | 21 - 32 |  |
| cmdPDF\_Click | 34 - 55 | '+ ' Procedure : cmdPDF\_Click ' Comments : display report in preview mode ' Parameters: - ' Modified : 01 Jul 1998 JSL ' 11 Jul 2003 LBlanken '- |
| cmdPreview\_Click | 64 - 79 |  |
| cmdPrint\_Click | 88 - 103 |  |
| cmdShowHide\_Click | 172 - 203 |  |
| DisplayReport | 112 - 164 |  |
| Form\_Open | 212 - 234 |  |
| getEndYear | 262 - 295 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetText | 243 - 260 |  |
| frmRptProcurementTable | (General Declarations) | 1 - 31 | '+ ' Module : Form\_frmRptProcurementTable ' Description: Code for Procurement Report parameter form ' Procedures : cboDisplay\_AfterUpdate() ' cboUseMonths\_AfterUpdate() ' cboYear\_AfterUpdate() ' cmdPreview\_Click() ' cmdPrint\_Click() ' cmdShowHide\_Click() ' DisplayReport(intView As Integer) ' Form\_Load() ' Form\_Open(Cancel As Integer) ' getValidMonths(strID As String) ' ini\_cboProduct() ' ini\_cboYear() ' lstDisplay1\_AfterUpdate() ' SetMinMax() ' SetText() ' UpdateControls(lngColumn As Long) |
| cboDisplay\_AfterUpdate | 38 - 100 | '+ ' Procedure : cboDisplay\_AfterUpdate ' Comments : updates listbox based on category selected ' Parameters: - ' Modified : 11 Jul 2003 LBlanken '- |
| cboUseMonths\_AfterUpdate | 108 - 124 |  |
| cboYear\_AfterUpdate | 133 - 162 |  |
| cmdPDF\_Click | 164 - 192 | ' Procedure : cmdPDF\_Click ' Comments : display report in preview mode ' Parameters: - ' Modified : 1 Aug 2006 LBlanken '- |
| cmdPreview\_Click | 201 - 225 |  |
| cmdPrint\_Click | 233 - 254 |  |
| cmdShowHide\_Click | 262 - 304 |  |
| DisplayReport | 313 - 369 |  |
| Form\_Error | 379 - 409 |  |
| Form\_Load | 418 - 431 |  |
| Form\_Open | 440 - 531 |  |
| getValidMonths | 539 - 593 |  |
| ini\_cboProduct | 603 - 658 |  |
| ini\_cboYear | 668 - 709 |  |
| lstDisplay1\_AfterUpdate | 719 - 816 |  |
| lstDisplay1\_KeyDown | 951 - 953 |  |
| lstDisplay1\_KeyUp | 955 - 957 |  |
| SetMinMax | 824 - 861 |  |
| SetText | 870 - 889 |  |
| ShowDOM | 940 - 949 |  |
| ShowFirstSelected | 959 - 1000 |  |
| UpdateControls | 898 - 939 |  |
| frmRptShipmentOrders | (General Declarations) | 1 - 35 | '+ ' Module : Form\_frmRptShipmentOrders ' Description: Code for Shipment Order Report parameter form ' Procedures : cboFrom\_AfterUpdate() ' cboThrough\_AfterUpdate() ' cmdPreview\_Click() ' cmdPrint\_Click() ' cmdShowHide\_Click() ' CreateShipQuery() ' DisplayReport(lngMode As Long) ' FillFilterCombos(strType As String) ' Form\_Load() ' lstDisplay1\_AfterUpdate() ' lstDisplay2\_AfterUpdate() ' MakeFilters() ' SetListSource() ' SetPageBreak(strReport As String, intMode As Integer) ' SetText() ' VerifyDateRange() |
| cboDisplay\_AfterUpdate | 36 - 79 |  |
| cboFrom\_AfterUpdate | 88 - 112 |  |
| cboPeriod\_AfterUpdate | 114 - 137 |  |
| cboPeriod\_GotFocus | 141 - 143 |  |
| cboThrough\_AfterUpdate | 152 - 170 |  |
| cmdPDF\_Click | 172 - 202 | '+ ' Procedure : cmdPDF\_Click ' Comments : display report in preview mode ' Parameters: - ' Modified : 1 AUG 2006 LBlanken '- |
| cmdPreview\_Click | 211 - 237 |  |
| cmdPrint\_Click | 246 - 272 |  |
| cmdShowHide\_Click | 280 - 323 |  |
| CreateShipQuery | 334 - 372 |  |
| DisplayReport | 381 - 434 |  |
| FillFilterCombos | 444 - 507 |  |
| Form\_Error | 517 - 547 |  |
| Form\_Load | 558 - 603 |  |
| Form\_Open | 611 - 649 |  |
| lstDisplay1\_AfterUpdate | 658 - 691 |  |
| lstDisplay1\_KeyDown | 693 - 695 |  |
| lstDisplay1\_KeyUp | 697 - 699 |  |
| lstDisplay2\_AfterUpdate | 709 - 741 |  |
| lstDisplay2\_KeyDown | 1023 - 1025 |  |
| lstDisplay2\_KeyUp | 1027 - 1029 |  |
| MakeFilters | 751 - 846 |  |
| SetListSource | 855 - 893 |  |
| SetPageBreak | 903 - 932 |  |
| SetText | 941 - 966 |  |
| VerifyDateRange | 976 - 1021 |  |
| frmRptShipmentSummary | (General Declarations) | 1 - 36 | '+ ' Module : Form\_frmRptShipmentSummary ' Description: code for shipment summary report parameter form ' Procedures : cboFrom\_AfterUpdate() ' cboThrough\_AfterUpdate() ' cmdPreview\_Click() ' cmdPrint\_Click() ' cmdShowHide\_Click() ' CreateShipQuery() ' DisplayReport(lngMode As Long) ' FillFilterCombos(strType As String) ' Form\_Load() ' Form\_Open(Cancel As Integer) ' lstDisplay1\_AfterUpdate() ' lstDisplay2\_AfterUpdate() ' optDisplay\_AfterUpdate() ' SetListSource() ' SetPageBreak(strReport As String, intMode As Integer) ' SetText() ' VerifyDateRange() |
| cboDisplay\_AfterUpdate | 37 - 93 |  |
| cboFrom\_AfterUpdate | 102 - 141 |  |
| cboPeriod\_AfterUpdate | 143 - 167 |  |
| cboPeriod\_GotFocus | 170 - 172 |  |
| cboThrough\_AfterUpdate | 181 - 199 |  |
| cmdPDF\_Click | 201 - 233 | '+ ' Procedure : cmdPDF\_Click ' Comments : display report in normal mode (i.e. send to printer) ' Parameters: - ' Modified : 01 AUG 2006 LKB '- |
| cmdPreview\_Click | 242 - 270 |  |
| cmdPrint\_Click | 279 - 306 |  |
| cmdShowHide\_Click | 314 - 361 |  |
| CreateShipQuery | 372 - 411 |  |
| DisplayReport | 420 - 471 |  |
| FillFilterCombos | 481 - 542 |  |
| Form\_Error | 552 - 582 |  |
| Form\_Load | 593 - 655 |  |
| Form\_Open | 666 - 714 |  |
| lstDisplay1\_AfterUpdate | 723 - 772 |  |
| lstDisplay1\_KeyDown | 774 - 776 |  |
| lstDisplay1\_KeyUp | 778 - 780 |  |
| lstDisplay2\_AfterUpdate | 789 - 821 |  |
| lstDisplay2\_KeyDown | 823 - 825 |  |
| lstDisplay2\_KeyUp | 827 - 829 |  |
| MakeFilters | 1046 - 1147 |  |
| optDisplay\_AfterUpdate | 837 - 856 |  |
| SetListSource | 865 - 902 |  |
| SetPageBreak | 912 - 941 |  |
| SetText | 950 - 981 |  |
| VerifyDateRange | 991 - 1036 |  |
| frmRptStockStatus | (General Declarations) | 1 - 36 | '+ ' Module : Form\_frmRptStockStatus ' Description: code for stock status report parameter form ' Procedures : SetListboxSelected(lstBox As ListBox, fSelected As Boolean) ' cboDisplay\_AfterUpdate() ' cboFrom\_AfterUpdate() ' cboGrouping\_AfterUpdate() ' cboThrough\_AfterUpdate() ' chkAllSupplier\_AfterUpdate() ' cmdPreview\_Click() ' cmdPrint\_Click() ' cmdShowHide\_Click() ' DisplayReport(intView As Integer) ' Form\_Load() ' Form\_Open(Cancel As Integer) ' ini\_cboProduct() ' ini\_cboYears() ' lstDisplay1\_AfterUpdate() ' lstDisplay2\_AfterUpdate() ' optDisplay\_AfterUpdate() ' SetText() ' Update\_Report() ' UpdateControls(lngColumn As Long) |
| cboDisplay\_AfterUpdate | 73 - 92 |  |
| cboFrom\_AfterUpdate | 103 - 130 |  |
| cboGrouping\_AfterUpdate | 139 - 165 |  |
| cboPeriod\_AfterUpdate | 166 - 188 |  |
| cboPeriod\_GotFocus | 191 - 193 |  |
| cboThrough\_AfterUpdate | 202 - 228 |  |
| chkAllSupplier\_AfterUpdate | 236 - 266 |  |
| cmdAMC\_Click | 268 - 280 |  |
| cmdPDF\_Click | 282 - 309 | '+ ' Procedure : cmdPDF\_Click ' Comments : display report as PDF ' Parameters: - ' Modified : 01 AUG 2006 LKB '- |
| cmdPreview\_Click | 318 - 343 |  |
| cmdPrint\_Click | 352 - 374 |  |
| cmdShowHide\_Click | 384 - 411 |  |
| DisplayReport | 422 - 607 |  |
| Form\_Error | 621 - 651 |  |
| Form\_Load | 660 - 676 |  |
| Form\_Open | 685 - 804 |  |
| GetSourceObject | 813 - 853 |  |
| ini\_cboProduct | 863 - 919 |  |
| ini\_cboYears | 930 - 1012 |  |
| lstDisplay1\_AfterUpdate | 1021 - 1247 |  |
| lstDisplay1\_KeyDown | 1250 - 1252 |  |
| lstDisplay1\_KeyUp | 1254 - 1256 |  |
| lstDisplay2\_AfterUpdate | 1264 - 1283 |  |
| lstDisplay2\_KeyDown | 1285 - 1287 |  |
| lstDisplay2\_KeyUp | 1289 - 1291 |  |
| optDisplay\_AfterUpdate | 1299 - 1436 |  |
| setFormForMatrixReport | 1579 - 1597 |  |
| SetListboxSelected | 44 - 65 | '+ ' Procedure : SetListboxSelected ' Comments : sets all items in listbox to true or false ' Parameters: lstBox - listbox to update ' fSelected - true to select all, false to deselect ' Modified : 11 Jul 2003 LBlanken '- |
| SetText | 1445 - 1464 |  |
| ShowFirstSelected | 1516 - 1570 |  |
| UpdateControls | 1473 - 1514 |  |
| frmRptStockStatusDateRange | (General Declarations) | 1 - 35 | '+ ' Module : Form\_frmRptStockStatus ' Description: code for stock status report parameter form ' Procedures : SetListboxSelected(lstBox As ListBox, fSelected As Boolean) ' cboDisplay\_AfterUpdate() ' cboFrom\_AfterUpdate() ' cboGrouping\_AfterUpdate() ' cboThrough\_AfterUpdate() ' chkAllSupplier\_AfterUpdate() ' cmdPreview\_Click() ' cmdPrint\_Click() ' cmdShowHide\_Click() ' DisplayReport(intView As Integer) ' Form\_Load() ' Form\_Open(Cancel As Integer) ' ini\_cboProduct() ' ini\_cboYears() ' lstDisplay1\_AfterUpdate() ' lstDisplay2\_AfterUpdate() ' optDisplay\_AfterUpdate() ' SetText() ' Update\_Report() ' UpdateControls(lngColumn As Long) |
| cboDisplay\_AfterUpdate | 72 - 91 |  |
| cboFrom\_AfterUpdate | 100 - 127 |  |
| cboGrouping\_AfterUpdate | 136 - 162 |  |
| cboThrough\_AfterUpdate | 171 - 197 |  |
| chkAllSupplier\_AfterUpdate | 205 - 235 |  |
| cmdAMC\_Click | 237 - 244 |  |
| cmdPDF\_Click | 246 - 273 | '+ ' Procedure : cmdPDF\_Click ' Comments : display report as PDF ' Parameters: - ' Modified : 01 AUG 2006 LKB '- |
| cmdPreview\_Click | 282 - 305 |  |
| cmdPrint\_Click | 314 - 336 |  |
| cmdShowHide\_Click | 344 - 367 |  |
| DisplayReport | 376 - 474 |  |
| Form\_Error | 484 - 514 |  |
| Form\_Load | 523 - 539 |  |
| Form\_Open | 548 - 657 |  |
| GetSourceObject | 666 - 698 |  |
| ini\_cboProduct | 708 - 764 |  |
| ini\_cboYears | 775 - 857 |  |
| lstDisplay1\_AfterUpdate | 866 - 1014 |  |
| lstDisplay1\_KeyDown | 1017 - 1019 |  |
| lstDisplay1\_KeyUp | 1021 - 1023 |  |
| lstDisplay2\_AfterUpdate | 1031 - 1050 |  |
| lstDisplay2\_KeyDown | 1052 - 1054 |  |
| lstDisplay2\_KeyUp | 1056 - 1058 |  |
| optDisplay\_AfterUpdate | 1066 - 1203 |  |
| SetListboxSelected | 43 - 64 | '+ ' Procedure : SetListboxSelected ' Comments : sets all items in listbox to true or false ' Parameters: lstBox - listbox to update ' fSelected - true to select all, false to deselect ' Modified : 11 Jul 2003 LBlanken '- |
| SetText | 1212 - 1231 |  |
| ShowFirstSelected | 1283 - 1337 |  |
| UpdateControls | 1240 - 1281 |  |
| frmRptSupplyPlan | (General Declarations) | 1 - 35 | '+ ' Module : Form\_frmRptStockStatus ' Description: code for stock status report parameter form ' Procedures : SetListboxSelected(lstBox As ListBox, fSelected As Boolean) ' cboDisplay\_AfterUpdate() ' cboFrom\_AfterUpdate() ' cboGrouping\_AfterUpdate() ' cboThrough\_AfterUpdate() ' chkAllSupplier\_AfterUpdate() ' cmdPreview\_Click() ' cmdPrint\_Click() ' cmdShowHide\_Click() ' DisplayReport(intView As Integer) ' Form\_Load() ' Form\_Open(Cancel As Integer) ' ini\_cboProduct() ' ini\_cboYears() ' lstDisplay1\_AfterUpdate() ' lstDisplay2\_AfterUpdate() ' optDisplay\_AfterUpdate() ' SetText() ' Update\_Report() ' UpdateControls(lngColumn As Long) |
| cboDisplay\_AfterUpdate | 72 - 91 |  |
| cboFrom\_AfterUpdate | 100 - 128 |  |
| cboGrouping\_AfterUpdate | 137 - 163 |  |
| cboPeriod\_AfterUpdate | 165 - 188 |  |
| cboPeriod\_GotFocus | 191 - 193 |  |
| cboThrough\_AfterUpdate | 202 - 228 |  |
| chkAllSupplier\_AfterUpdate | 236 - 266 |  |
| chkRoundUp\_AfterUpdate | 268 - 313 |  |
| cmdPDF\_Click | 318 - 345 | '+ ' Procedure : cmdPDF\_Click ' Comments : display report as PDF ' Parameters: - ' Modified : 01 AUG 2006 LKB '- |
| cmdPreview\_Click | 354 - 377 |  |
| cmdPrint\_Click | 386 - 408 |  |
| cmdShowHide\_Click | 416 - 439 |  |
| DisplayReport | 448 - 509 |  |
| Form\_Error | 519 - 549 |  |
| Form\_Load | 558 - 574 |  |
| Form\_Open | 583 - 695 |  |
| GetSourceObject | 704 - 721 |  |
| ini\_cboProduct | 731 - 786 |  |
| ini\_cboYears | 797 - 871 |  |
| lstDisplay1\_AfterUpdate | 880 - 1008 |  |
| lstDisplay1\_KeyDown | 1011 - 1013 |  |
| lstDisplay1\_KeyUp | 1015 - 1017 |  |
| lstDisplay2\_AfterUpdate | 1025 - 1044 |  |
| lstDisplay2\_KeyDown | 1046 - 1048 |  |
| lstDisplay2\_KeyUp | 1050 - 1052 |  |
| optDisplay\_AfterUpdate | 1060 - 1146 |  |
| SetListboxSelected | 43 - 64 | '+ ' Procedure : SetListboxSelected ' Comments : sets all items in listbox to true or false ' Parameters: lstBox - listbox to update ' fSelected - true to select all, false to deselect ' Modified : 11 Jul 2003 LBlanken '- |
| SetText | 1155 - 1174 |  |
| ShowFirstSelected | 1226 - 1272 |  |
| UpdateControls | 1183 - 1224 |  |
| frmShipments | (General Declarations) | 1 - 52 | '+ ' Module : Form\_frmShipments ' Description: Allows user to input shipments. When form ' opens, it finds the record in the list box that is closest ' to the current date. The user then has the option to edit ' delete, or add a record. ' Procedures : MovetoClosestShipment() ' CalculateUnitCost() ' cboSelect1\_AfterUpdate() ' cboSelect1\_GotFocus() ' cboSelect2\_AfterUpdate() ' cboSelect2\_GotFocus() ' cboSupplier\_AfterUpdate() ' cmdCopy\_Click() ' cmdDelete\_Click() ' cmdNew\_Click() ' cmdSave\_Click() ' Date\_Consist() ' Estimate\_LeadTimes(iEvent As Integer) ' Form\_Activate() ' Form\_AfterInsert() ' Form\_BeforeInsert(Cancel As Integer) ' Form\_BeforeUpdate(Cancel As Integer) ' Form\_Delete(Cancel As Integer) ' Form\_Dirty(Cancel As Integer) ' Form\_Error(DataErr As Integer, Response As Integer) ' Form\_Open(Cancel As Integer) ' GetCaseLot() ' getEstFreightCost() ' HasValidPricing() ' lstSelect\_Click() ' SetText() ' txtActOrdered\_AfterUpdate() ' txtActOrdered\_BeforeUpdate(Cancel As Integer) ' txtActShipped\_AfterUpdate() ' txtActShipped\_BeforeUpdate(Cancel As Integer) ' txtDate\_AfterUpdate() ' txtDate\_BeforeUpdate(Cancel As Integer) ' txtQuantity\_AfterUpdate() |
| CalculateUnitCost | 136 - 212 |  |
| cboSelect1\_AfterUpdate | 220 - 270 |  |
| cboSelect1\_GotFocus | 278 - 297 |  |
| cboSelect2\_AfterUpdate | 306 - 343 |  |
| cboSelect2\_GotFocus | 353 - 372 |  |
| cboStatus\_AfterUpdate | 374 - 417 |  |
| cboSupplier\_AfterUpdate | 425 - 441 |  |
| cmdBack\_Click | 444 - 447 |  |
| cmdCopy\_Click | 457 - 501 |  |
| cmdDelete\_Click | 510 - 576 |  |
| cmdNew\_Click | 585 - 618 |  |
| cmdPlan\_Click | 621 - 646 |  |
| cmdSave\_Click | 654 - 702 |  |
| cmdSearch\_Click | 929 - 936 |  |
| Date\_Consist | 712 - 824 |  |
| Estimate\_LeadTimes | 833 - 927 |  |
| Form\_Activate | 945 - 958 |  |
| Form\_BeforeInsert | 966 - 1014 |  |
| Form\_BeforeUpdate | 1023 - 1090 |  |
| Form\_Dirty | 1099 - 1114 |  |
| Form\_Error | 1124 - 1168 |  |
| Form\_Open | 1177 - 1334 |  |
| GetCaseLot | 1757 - 1854 |  |
| getEstFreightCost | 1345 - 1394 |  |
| HasValidPricing | 1407 - 1462 |  |
| lstSelect\_Click | 1470 - 1497 |  |
| MovetoClosestShipment | 60 - 126 | '+ ' Procedure : MovetoClosestShipment ' Comments : Find the record closest to todays date. ' Parameters: - ' Created : 03 Feb 2000 Jeff Leiner ' Modified : 23 Jun 2003 LBlanken '- |
| SetText | 1506 - 1520 |  |
| txtActOrdered\_AfterUpdate | 1529 - 1550 |  |
| txtActOrdered\_BeforeUpdate | 1559 - 1580 |  |
| txtActShipped\_AfterUpdate | 1589 - 1614 |  |
| txtActShipped\_BeforeUpdate | 1623 - 1644 |  |
| txtDate\_AfterUpdate | 1652 - 1670 |  |
| txtDate\_BeforeUpdate | 1679 - 1709 |  |
| txtQuantity\_AfterUpdate | 1719 - 1748 |  |
| frmSplash | (General Declarations) | 1 - 20 | ' Module : Form\_frmSplash ' Description: ' Procedures : cboLanguage\_AfterUpdate() ' cmdCancel\_Click() ' cmdContinue\_Click() ' Form\_Open(Cancel As Integer) ' Form\_Timer() ' optUpgradeChoice\_AfterUpdate() ' SetText() |
| cboLanguage\_AfterUpdate | 21 - 42 | ' Comments : When the language is modified, update the system ' : variables, and the display. ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Form\_Open | 44 - 109 | ' Comments : On Open see if an upgrade is needed. If yes ' : give the user the option, if not act as a splash ' : screen. ' Parameters : ' Returns : ' Created : 06/29/98 JSL ' Modified : ' ' -------------------------------------------------------- |
| Form\_Timer | 111 - 143 | ' Comments : if TimerInterval > 0 then count the iterations and ' : do do the correct events ' Parameters : ' Returns : ' Created : 06/29/98 JSL ' Modified : ' ' -------------------------------------------------------- |
| SetText | 145 - 176 | ' Comments : Display the text in the proper language ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| frmStock | (General Declarations) | 1 - 42 | '+ ' Module : Form\_frmStock ' Description: Allows user to input inventory adjustments. When form ' opens, it finds the record in the list box that is closest ' to the current date. The user then has the option to edit ' delete, or add a record. ' Procedures : FindClosestRecord() ' cboSelect1\_AfterUpdate() ' cboSelect1\_GotFocus() ' cboSelect2\_AfterUpdate() ' cboSelect2\_GotFocus() ' cmdCount\_Click() ' cmdDelete\_Click() ' cmdNew\_Click() ' cmdSave\_Click() ' Form\_Activate() ' Form\_BeforeInsert(Cancel As Integer) ' Form\_BeforeUpdate(Cancel As Integer) ' Form\_Current() ' Form\_Dirty(Cancel As Integer) ' Form\_Error(DataErr As Integer, Response As Integer) ' Form\_Open(Cancel As Integer) ' lstSelect\_Click() ' SetText() |
| cboSelect1\_AfterUpdate | 118 - 168 |  |
| cboSelect1\_GotFocus | 176 - 195 |  |
| cboSelect2\_AfterUpdate | 204 - 263 |  |
| cboSelect2\_GotFocus | 273 - 292 |  |
| cmdBack\_Click | 294 - 298 |  |
| cmdCount\_Click | 307 - 369 |  |
| cmdDelete\_Click | 379 - 431 |  |
| cmdNew\_Click | 441 - 483 |  |
| cmdSave\_Click | 492 - 547 |  |
| FindClosestRecord | 52 - 110 | '+ ' Procedure : FindClosestRecord ' Comments : Find the record in the list box that is ' closest to the current date ' Parameters: - ' Created : 04 Feb 2000 Jeff Leiner ' Modified : 03 Jun 2003 LBlanken ' 16 Jun 2003 LBlanken '- |
| Form\_Activate | 556 - 570 |  |
| Form\_BeforeInsert | 579 - 604 |  |
| Form\_BeforeUpdate | 613 - 662 |  |
| Form\_Current | 671 - 699 |  |
| Form\_Dirty | 708 - 723 |  |
| Form\_Error | 734 - 785 |  |
| Form\_Open | 795 - 953 |  |
| lstSelect\_Click | 962 - 996 |  |
| SetText | 1006 - 1020 |  |
| txtAmount\_GotFocus | 1022 - 1025 |  |
| txtDate\_AfterUpdate | 1031 - 1034 | 'mtracy 4/5/07 bugzilla 10924 |
| txtDate\_GotFocus | 1027 - 1029 |  |
| frmSuppliers | (General Declarations) | 1 - 28 | '+ ' Module : Form\_frmSuppliers ' Description: Allows users to maintain the list of suppliers used ' throughout the system. ' Procedures : cmdDelete\_Click() ' cmdNew\_Click() ' cmdSave\_Click() ' Form\_Activate() ' Form\_BeforeUpdate(Cancel As Integer) ' Form\_Dirty(Cancel As Integer) ' Form\_Error(DataErr As Integer, Response As Integer) ' Form\_Open(Cancel As Integer) ' lstSelect\_Click() ' SetText() ' txtName\_AfterUpdate() |
| cmdDelete\_Click | 37 - 93 | '+ ' Procedure : cmdDelete\_Click ' Comments : deletes the selected item in listbox ' Parameters: - ' Modified : 01 Jul 1998 JSL ' 03 Jun 2003 LBlanken ' 16 Jun 2003 LBlanken '- |
| cmdNew\_Click | 103 - 135 |  |
| cmdSave\_Click | 144 - 206 |  |
| Form\_Activate | 215 - 230 |  |
| Form\_BeforeUpdate | 239 - 280 |  |
| Form\_Dirty | 289 - 304 |  |
| Form\_Error | 314 - 352 |  |
| Form\_Open | 363 - 429 |  |
| lstSelect\_Click | 439 - 466 |  |
| SetText | 476 - 490 |  |
| txtName\_AfterUpdate | 499 - 516 |  |
| UpdateControls | 518 - 530 |  |
| frmTypes | (General Declarations) | 1 - 4 |  |
| Form\_Open | 5 - 13 |  |
| lstMoveSelect1\_afterupdate | 37 - 39 |  |
| SetText | 21 - 35 |  |
| fsfrPipelineAction | (General Declarations) | 1 - 19 | '+ ' Module : Form\_fsfrPipelineAction ' Description: ' Procedures : Form\_Activate() ' Form\_Open(Cancel As Integer) ' getEndYear() ' SetText() ' txtProduct\_DblClick(Cancel As Integer) ' txtProduct\_KeyDown(KeyCode As Integer, Shift As Integer) |
| Form\_Activate | 27 - 96 | '+ ' Procedure : Form\_Activate ' Comments : move to correct record in table ' Parameters: - ' Modified : 01 Jul 1998 JSL ' 31 Jul 2003 LBlanken '- |
| Form\_Open | 105 - 144 |  |
| getEndYear | 154 - 186 |  |
| SetText | 195 - 208 |  |
| txtProduct\_DblClick | 217 - 246 |  |
| txtProduct\_KeyDown | 255 - 270 |  |
| fsfrPipelineProblem | (General Declarations) | 1 - 16 | ' Module : Form\_fsfrPipelineProblem ' Description: ' Procedures : Form\_Activate() ' Form\_Open(Cancel As Integer) ' getEndYear() ' SetText() ' txtProduct\_DblClick(Cancel As Integer) |
| Form\_Activate | 17 - 81 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Form\_Open | 83 - 120 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| getEndYear | 122 - 156 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetText | 158 - 177 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| txtProduct\_DblClick | 179 - 209 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| txtProduct\_KeyDown | 211 - 215 |  |
| fsfrProcurementTable | (General Declarations) | 1 - 16 | '+ ' Module : Form\_fsfrProcurementTable ' Description: ' Procedures : Form\_Open(Cancel As Integer) ' Form\_Current() ' setProductID() ' SetText() ' setYears() |
| Form\_Current | 48 - 62 |  |
| Form\_Open | 24 - 39 | '+ ' Procedure : Form\_Open ' Comments : set text, years and product ' Parameters: Cancel - ' Modified : 01 Jul 1998 JSL ' 28 Jul 2003 LBlanken '- |
| setProductID | 71 - 92 |  |
| SetText | 101 - 114 |  |
| SetYears | 123 - 143 |  |
| fsfrShipmentFunding | (General Declarations) | 1 - 13 | '+ ' Module : Form\_fsfrShipmentSupplier ' Description: ' Procedures : Form\_Open(Cancel As Integer) ' SetText() |
| Form\_Open | 21 - 34 | '+ ' Procedure : Form\_Open ' Comments : set text on form ' Parameters: Cancel - ' Modified : 01 Jul 1998 JSL ' 31 Jul 2003 LBlanken '- |
| SetText | 43 - 56 |  |
| fsfrShipmentOrder | (General Declarations) | 1 - 13 | '+ ' Module : Form\_fsfrShipmentOrder ' Description: ' Procedures : Form\_Open(Cancel As Integer) ' SetText() |
| Form\_Open | 21 - 34 | '+ ' Procedure : Form\_Open ' Comments : set text ' Parameters: Cancel - ' Modified : 01 Jul 1998 JSL ' 31 Jul 2003 LBlanken '- |
| SetText | 43 - 56 |  |
| fsfrShipmentProduct | (General Declarations) | 1 - 13 | '+ ' Module : Form\_fsfrShipmentProduct ' Description: ' Procedures : Form\_Open(Cancel As Integer) ' SetText() |
| Form\_Open | 21 - 34 | '+ ' Procedure : Form\_Open ' Comments : set text ' Parameters: Cancel - ' Modified : 01 Jul 1998 JSL ' 31 Jul 2003 LBlanken '- |
| SetText | 43 - 56 |  |
| fsfrShipmentSupplier | (General Declarations) | 1 - 13 | '+ ' Module : Form\_fsfrShipmentSupplier ' Description: ' Procedures : Form\_Open(Cancel As Integer) ' SetText() |
| Form\_Open | 21 - 34 | '+ ' Procedure : Form\_Open ' Comments : set text on form ' Parameters: Cancel - ' Modified : 01 Jul 1998 JSL ' 31 Jul 2003 LBlanken '- |
| SetText | 43 - 56 |  |
| fsfrStatus\_mon | (General Declarations) | 1 - 28 | '+ ' Module : Form\_fsfrStatus\_mon ' Description: ' Procedures : chkActFore\_DblClick(Cancel As Integer) ' chkActFore\_KeyDown(KeyCode As Integer, Shift As Integer) ' Form\_Open(Cancel As Integer) ' SetText() ' txtActFore\_DblClick(Cancel As Integer) ' txtBegBalance\_DblClick(Cancel As Integer) ' txtBegBalance\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtConsumption\_DblClick(Cancel As Integer) ' txtConsumption\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtQuantity\_DblClick(Cancel As Integer) ' txtQuantity\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtStatus\_DblClick(Cancel As Integer) ' txtStatus\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtStockAdj\_DblClick(Cancel As Integer) ' txtStockAdj\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtSupplier\_DblClick(Cancel As Integer) ' txtSupplier\_KeyDown(KeyCode As Integer, Shift As Integer) |
| chkActFore\_DblClick | 36 - 52 | '+ ' Procedure : chkActFore\_DblClick ' Comments : set globals and open consumption form ' Parameters: Cancel - ' Modified : 01 Jul 1998 JSL ' 12 Aug 2003 LBlanken '- |
| chkActFore\_KeyDown | 61 - 76 |  |
| Form\_Open | 85 - 98 |  |
| SetText | 107 - 120 |  |
| txtActFore\_DblClick | 129 - 145 |  |
| txtBegBalance\_DblClick | 154 - 169 |  |
| txtBegBalance\_KeyDown | 178 - 193 |  |
| txtConsumption\_DblClick | 202 - 219 |  |
| txtConsumption\_KeyDown | 228 - 243 |  |
| txtFunding\_DblClick | 245 - 270 |  |
| txtFunding\_KeyDown | 273 - 288 |  |
| txtQuantity\_DblClick | 298 - 322 |  |
| txtQuantity\_KeyDown | 331 - 346 |  |
| txtStatus\_DblClick | 355 - 379 |  |
| txtStatus\_KeyDown | 388 - 403 |  |
| txtStockAdj\_DblClick | 412 - 428 |  |
| txtStockAdj\_KeyDown | 437 - 452 |  |
| txtSupplier\_DblClick | 461 - 485 |  |
| txtSupplier\_KeyDown | 494 - 509 |  |
| fsfrStatus\_qtr | (General Declarations) | 1 - 28 | '+ ' Module : Form\_fsfrStatus\_qtr ' Description: ' Procedures : chkActFore\_DblClick(Cancel As Integer) ' chkActFore\_KeyDown(KeyCode As Integer, Shift As Integer) ' Form\_Open(Cancel As Integer) ' SetText() ' txtActFore\_DblClick(Cancel As Integer) ' txtBegBalance\_DblClick(Cancel As Integer) ' txtBegBalance\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtConsumption\_DblClick(Cancel As Integer) ' txtConsumption\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtQuantity\_DblClick(Cancel As Integer) ' txtQuantity\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtStatus\_DblClick(Cancel As Integer) ' txtStatus\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtStockAdj\_DblClick(Cancel As Integer) ' txtStockAdj\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtSupplier\_DblClick(Cancel As Integer) ' txtSupplier\_KeyDown(KeyCode As Integer, Shift As Integer) |
| chkActFore\_DblClick | 36 - 52 | '+ ' Procedure : chkActFore\_DblClick ' Comments : set globals and open consumption form ' Parameters: Cancel - ' Modified : 01 Jul 1998 JSL ' 12 Aug 2003 LBlanken '- |
| chkActFore\_KeyDown | 61 - 76 |  |
| Form\_Open | 85 - 98 |  |
| SetText | 107 - 120 |  |
| txtActFore\_DblClick | 129 - 145 |  |
| txtBegBalance\_DblClick | 154 - 169 |  |
| txtBegBalance\_KeyDown | 178 - 194 |  |
| txtConsumption\_DblClick | 203 - 219 |  |
| txtConsumption\_KeyDown | 228 - 243 |  |
| txtQuantity\_DblClick | 252 - 277 |  |
| txtQuantity\_KeyDown | 286 - 301 |  |
| txtStatus\_DblClick | 310 - 335 |  |
| txtStatus\_KeyDown | 344 - 359 |  |
| txtStockAdj\_DblClick | 368 - 384 |  |
| txtStockAdj\_KeyDown | 393 - 408 |  |
| txtSupplier\_DblClick | 417 - 442 |  |
| txtSupplier\_KeyDown | 451 - 466 |  |
| fsfrStatusMethod | (General Declarations) | 1 - 29 | '+ ' Module : Form\_fsfrStatus\_mon ' Description: ' Procedures : chkActFore\_DblClick(Cancel As Integer) ' chkActFore\_KeyDown(KeyCode As Integer, Shift As Integer) ' Form\_Open(Cancel As Integer) ' SetText() ' txtActFore\_DblClick(Cancel As Integer) ' txtBegBalance\_DblClick(Cancel As Integer) ' txtBegBalance\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtConsumption\_DblClick(Cancel As Integer) ' txtConsumption\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtQuantity\_DblClick(Cancel As Integer) ' txtQuantity\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtStatus\_DblClick(Cancel As Integer) ' txtStatus\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtStockAdj\_DblClick(Cancel As Integer) ' txtStockAdj\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtSupplier\_DblClick(Cancel As Integer) ' txtSupplier\_KeyDown(KeyCode As Integer, Shift As Integer) |
| Form\_Open | 37 - 50 | '+ ' Procedure : Form\_Open ' Comments : set text on report ' Parameters: Cancel - ' Modified : 01 Jul 1998 JSL ' 12 Aug 2003 LBlanken '- |
| SetText | 59 - 72 |  |
| fsfrStatusMethod\_mon | (General Declarations) | 1 - 28 | '+ ' Module : Form\_fsfrStatusMethod\_mon ' Description: ' Procedures : chkActFore\_DblClick(Cancel As Integer) ' chkActFore\_KeyDown(KeyCode As Integer, Shift As Integer) ' Form\_Open(Cancel As Integer) ' SetText() ' txtActFore\_DblClick(Cancel As Integer) ' txtBegBalance\_DblClick(Cancel As Integer) ' txtBegBalance\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtConsumption\_DblClick(Cancel As Integer) ' txtConsumption\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtQuantity\_DblClick(Cancel As Integer) ' txtQuantity\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtStatus\_DblClick(Cancel As Integer) ' txtStatus\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtStockAdj\_DblClick(Cancel As Integer) ' txtStockAdj\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtSupplier\_DblClick(Cancel As Integer) ' txtSupplier\_KeyDown(KeyCode As Integer, Shift As Integer) |
| chkActFore\_DblClick | 36 - 52 | '+ ' Procedure : chkActFore\_DblClick ' Comments : set globals and open consumption form ' Parameters: Cancel - ' Modified : 01 Jul 1998 JSL ' 12 Aug 2003 LBlanken '- |
| chkActFore\_KeyDown | 61 - 76 |  |
| Form\_Open | 85 - 98 |  |
| SetText | 107 - 120 |  |
| txtActFore\_DblClick | 129 - 145 |  |
| txtBegBalance\_DblClick | 154 - 169 |  |
| txtBegBalance\_KeyDown | 178 - 193 |  |
| txtConsumption\_DblClick | 202 - 218 |  |
| txtConsumption\_KeyDown | 227 - 242 |  |
| txtQuantity\_DblClick | 251 - 275 |  |
| txtQuantity\_KeyDown | 284 - 299 |  |
| txtStatus\_DblClick | 308 - 332 |  |
| txtStatus\_KeyDown | 341 - 356 |  |
| txtStockAdj\_DblClick | 365 - 381 |  |
| txtStockAdj\_KeyDown | 390 - 405 |  |
| txtSupplier\_DblClick | 414 - 438 |  |
| txtSupplier\_KeyDown | 447 - 462 |  |
| fsfrStatusMethod\_qtr | (General Declarations) | 1 - 28 | '+ ' Module : Form\_fsfrStatusMethod\_qtr ' Description: ' Procedures : chkActFore\_DblClick(Cancel As Integer) ' chkActFore\_KeyDown(KeyCode As Integer, Shift As Integer) ' Form\_Open(Cancel As Integer) ' SetText() ' txtActFore\_DblClick(Cancel As Integer) ' txtBegBalance\_DblClick(Cancel As Integer) ' txtBegBalance\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtConsumption\_DblClick(Cancel As Integer) ' txtConsumption\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtQuantity\_DblClick(Cancel As Integer) ' txtQuantity\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtStatus\_DblClick(Cancel As Integer) ' txtStatus\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtStockAdj\_DblClick(Cancel As Integer) ' txtStockAdj\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtSupplier\_DblClick(Cancel As Integer) ' txtSupplier\_KeyDown(KeyCode As Integer, Shift As Integer) |
| chkActFore\_DblClick | 36 - 52 | '+ ' Procedure : chkActFore\_DblClick ' Comments : set globals and open consumption form ' Parameters: Cancel - ' Modified : 01 Jul 1998 JSL ' 12 Aug 2003 LBlanken '- |
| chkActFore\_KeyDown | 61 - 76 |  |
| Form\_Open | 85 - 98 |  |
| SetText | 107 - 120 |  |
| txtActFore\_DblClick | 129 - 145 |  |
| txtBegBalance\_DblClick | 154 - 169 |  |
| txtBegBalance\_KeyDown | 178 - 193 |  |
| txtConsumption\_DblClick | 202 - 218 |  |
| txtConsumption\_KeyDown | 227 - 242 |  |
| txtQuantity\_DblClick | 251 - 275 |  |
| txtQuantity\_KeyDown | 284 - 299 |  |
| txtStatus\_DblClick | 308 - 332 |  |
| txtStatus\_KeyDown | 341 - 356 |  |
| txtStockAdj\_DblClick | 365 - 381 |  |
| txtStockAdj\_KeyDown | 390 - 405 |  |
| txtSupplier\_DblClick | 414 - 438 |  |
| txtSupplier\_KeyDown | 447 - 462 |  |
| fsfrStatusProduct | (General Declarations) | 1 - 29 | '+ ' Module : Form\_fsfrStatus\_mon ' Description: ' Procedures : chkActFore\_DblClick(Cancel As Integer) ' chkActFore\_KeyDown(KeyCode As Integer, Shift As Integer) ' Form\_Open(Cancel As Integer) ' SetText() ' txtActFore\_DblClick(Cancel As Integer) ' txtBegBalance\_DblClick(Cancel As Integer) ' txtBegBalance\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtConsumption\_DblClick(Cancel As Integer) ' txtConsumption\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtQuantity\_DblClick(Cancel As Integer) ' txtQuantity\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtStatus\_DblClick(Cancel As Integer) ' txtStatus\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtStockAdj\_DblClick(Cancel As Integer) ' txtStockAdj\_KeyDown(KeyCode As Integer, Shift As Integer) ' txtSupplier\_DblClick(Cancel As Integer) ' txtSupplier\_KeyDown(KeyCode As Integer, Shift As Integer) |
| Form\_Open | 37 - 50 | '+ ' Procedure : Form\_Open ' Comments : set text on report ' Parameters: Cancel - ' Modified : 01 Jul 1998 JSL ' 12 Aug 2003 LBlanken '- |
| SetText | 59 - 72 |  |

Reports

Table - List of Reports

| ModuleName | Procedure | Lines | Comments |
| --- | --- | --- | --- |
| rptAMCResult | (General Declarations) | 1 - 1 |  |
| Report\_Close | 53 - 86 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- ' : June 26, 2009 mahmed ' enable on Application close button on close. '- |
| Report\_Open | 25 - 51 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetText | 9 - 23 | '+ ' Procedure : SetText ' Comments : Set text on report ' Parameters: - ' Modified : 01 Jul 1998 JSL ' 11 Jul 2003 LBlanken '- |
| rptAnnualCosts | (General Declarations) | 1 - 14 | ' Module : Report\_rptAnnualCosts ' Description: ' Procedures : GroupHeader5\_Print(Cancel As Integer, PrintCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail1\_Format | 15 - 22 |  |
| GroupHeader5\_Print | 24 - 58 | ' Comments : ' Parameters : Cancel ' PrintCount ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Report\_Close | 60 - 88 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- ' : June 26, 2009 mahmed ' enable on Application close button on close. '- |
| Report\_Open | 90 - 115 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetText | 117 - 187 | ' Comments : ' Parameters : ' Returns : ' Created : 06/16/98 JSL ' Modified : ' ' July 21, 2009 mahmed ' report subtitle now created from user-defined range ' -------------------------------------------------------- |
| rptAnnualCostsByFundingSource | (General Declarations) | 1 - 14 | ' Module : Report\_rptAnnualCosts ' Description: ' Procedures : GroupHeader5\_Print(Cancel As Integer, PrintCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail1\_Format | 15 - 22 |  |
| GroupHeader0\_Print | 24 - 62 | ' Comments : ' Parameters : Cancel ' PrintCount ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| GroupHeader5\_Print | 64 - 101 | ' Comments : ' Parameters : Cancel ' PrintCount ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Report\_Close | 103 - 131 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- ' : June 26, 2009 mahmed ' enable on Application close button on close. '- |
| Report\_Open | 133 - 158 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetText | 160 - 230 | ' Comments : ' Parameters : ' Returns : ' Created : 06/16/98 JSL ' Modified : ' ' July 21, 2009 mahmed ' report subtitle now created from user-defined range ' -------------------------------------------------------- |
| rptAnnualCostsBySupplier | (General Declarations) | 1 - 14 | ' Module : Report\_rptAnnualCosts ' Description: ' Procedures : GroupHeader5\_Print(Cancel As Integer, PrintCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail1\_Format | 15 - 22 |  |
| GroupHeader0\_Print | 24 - 62 | ' Comments : ' Parameters : Cancel ' PrintCount ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| GroupHeader5\_Print | 64 - 101 | ' Comments : ' Parameters : Cancel ' PrintCount ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Report\_Close | 103 - 131 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- ' : June 26, 2009 mahmed ' enable on Application close button on close. '- |
| Report\_Open | 133 - 161 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetText | 163 - 233 | ' Comments : ' Parameters : ' Returns : ' Created : 06/16/98 JSL ' Modified : ' ' July 21, 2009 mahmed ' report subtitle now created from user-defined range ' -------------------------------------------------------- |
| rptCloneConsumptionFinal | (General Declarations) | 1 - 4 |  |
| Report\_Close | 81 - 119 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- ' : June 26, 2009 mahmed ' enable on Application close button on close. '- |
| Report\_Open | 5 - 39 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetText | 41 - 79 | ' Comments : ' Parameters : ' Returns : ' Created : 06/16/98 JSL ' Modified : ' ' July 21, 2009 mahmed ' report subtitle now created from user-defined range ' -------------------------------------------------------- |
| rptCloneConsumptionPreview | (General Declarations) | 1 - 4 |  |
| Report\_Close | 80 - 116 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- ' : June 26, 2009 mahmed ' enable on Application close button on close. '- |
| Report\_Open | 5 - 38 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetText | 40 - 78 | ' Comments : ' Parameters : ' Returns : ' Created : 06/16/98 JSL ' Modified : ' ' July 21, 2009 mahmed ' report subtitle now created from user-defined range ' -------------------------------------------------------- |
| rptGraphConsumption | (General Declarations) | 1 - 31 | '+ ' Module : Report\_rptGraphConsumption ' Description: ' Procedures : PageHeader0\_Format(Cancel As Integer, intFormatCount As Integer) ' Report\_Close() ' Report\_NoData(Cancel As Integer) ' Report\_Open(Cancel As Integer) ' setRowSource() ' SetText() |
| PageHeader0\_Format | 39 - 85 | '+ ' Procedure : PageHeader0\_Format ' Comments : ' Parameters: Cancel ' intFormatCount - ' Modified : 11 Jul 2003 LBlanken '- |
| Report\_Close | 97 - 121 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| Report\_NoData | 130 - 150 | ' Comments : ' Parameters: Cancel - ' Modified : ' ' -------------------------------------------------- |
| Report\_Open | 159 - 217 | ' Comments : ' Parameters : cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| Report\_Page | 282 - 284 |  |
| setRowSource | 226 - 252 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetText | 261 - 280 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| rptGraphCYP | (General Declarations) | 1 - 16 | ' Module : Report\_rptGraphCYP ' Description: ' Procedures : GetProductNames() ' PageHeader0\_Format(Cancel As Integer, FormatCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' setRowSource() ' SetText() |
| GetProductNames | 17 - 61 | ' Comments : ' Parameters: - ' Returns : String - ' Modified : ' ' -------------------------------------------------- |
| PageHeader0\_Format | 63 - 73 | ' Comments : ' Parameters: Cancel ' FormatCount - ' Modified : ' ' -------------------------------------------------- |
| Report\_Close | 75 - 94 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- ' : June 26, 2009 mahmed ' enable on Application close button on close. '- |
| Report\_Open | 96 - 155 | ' Comments : ' Parameters : cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' July 21, 2009 mahmed ' report subtitle now created from user-defined range ' -------------------------------------------------------- |
| setRowSource | 157 - 183 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetText | 185 - 206 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| rptGraphStockStatus | (General Declarations) | 1 - 30 | '+ ' Module : Report\_rptGraphStockStatus ' Description: ' Procedures : FormHeader1\_Format(Cancel As Integer, intFormatCount As Integer) ' PageHeader0\_Format(Cancel As Integer, intFormatCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' setRowSource() ' SetText() |
| PageHeader0\_Format | 41 - 69 | '+ '+ ' Procedure : PageHeader0\_Format ' Comments : ' Parameters: Cancel ' intFormatCount - ' Modified : 11 Jul 2003 LBlanken ' July 21, 2009 mahmed ' report subtitle now created from user-defined range '- |
| Report\_Close | 83 - 108 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| Report\_Open | 117 - 174 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| setRowSource | 183 - 213 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetText | 222 - 241 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| rptGraphTrend | (General Declarations) | 1 - 16 | ' Module : Report\_rptGraphTrend ' Description: ' Procedures : Detail1\_Print(Cancel As Integer, PrintCount As Integer) ' GroupFooter5\_Print(Cancel As Integer, PrintCount As Integer) ' GroupFooter6\_Print(Cancel As Integer, PrintCount As Integer) ' Report\_Close() ' Report\_NoData(Cancel As Integer) ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail1\_Print | 17 - 28 | ' Comments : ' Parameters: Cancel ' PrintCount - ' Modified : ' ' -------------------------------------------------- |
| GroupFooter5\_Print | 30 - 41 | ' Comments : ' Parameters: Cancel ' PrintCount - ' Modified : ' ' -------------------------------------------------- |
| GroupFooter6\_Print | 43 - 54 | ' Comments : ' Parameters: Cancel ' PrintCount - ' Modified : ' ' -------------------------------------------------- |
| PageHeader0\_Format | 56 - 74 | 'mtracy 2/6/07 bugzilla 14399 |
| Report\_Close | 76 - 94 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- ' : June 26, 2009 mahmed ' enable on Application close button on close. '- |
| Report\_NoData | 96 - 105 | ' Comments : ' Parameters: Cancel - ' Modified : ' ' -------------------------------------------------- |
| Report\_Open | 107 - 211 | ' Comments : ' Parameters: Cancel - ' Modified : ' ' -------------------------------------------------- |
| SetText | 213 - 223 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| rptGraphTrendCons | (General Declarations) | 1 - 16 | ' Module : Report\_rptGraphTrendCons ' Description: ' Procedures : Detail1\_Print(Cancel As Integer, PrintCount As Integer) ' GroupFooter5\_Print(Cancel As Integer, PrintCount As Integer) ' GroupFooter6\_Print(Cancel As Integer, PrintCount As Integer) ' Report\_Close() ' Report\_NoData(Cancel As Integer) ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail1\_Print | 17 - 28 | ' Comments : ' Parameters: Cancel ' PrintCount - ' Modified : ' ' -------------------------------------------------- |
| GroupFooter5\_Print | 30 - 41 | ' Comments : ' Parameters: Cancel ' PrintCount - ' Modified : ' ' -------------------------------------------------- |
| GroupFooter6\_Print | 43 - 54 | ' Comments : ' Parameters: Cancel ' PrintCount - ' Modified : ' ' -------------------------------------------------- |
| PageHeader0\_Format | 56 - 74 | 'mtracy 2/6/07 bugzilla 14399 |
| Report\_Close | 76 - 94 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- ' : June 26, 2009 mahmed ' enable on Application close button on close. '- |
| Report\_NoData | 96 - 105 | ' Comments : ' Parameters: Cancel - ' Modified : ' ' -------------------------------------------------- |
| Report\_Open | 107 - 210 | ' Comments : ' Parameters: Cancel - ' Modified : ' ' -------------------------------------------------- |
| SetText | 212 - 222 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| rptImport | (General Declarations) | 1 - 26 | '+ ' Module : Report\_rptiMPORT ' Description: ' Procedures : GroupHeader1\_Format(Cancel As Integer, FormatCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Report\_Close | 27 - 49 |  |
| Report\_Open | 59 - 85 |  |
| SetText | 95 - 110 |  |
| rptInterpolate | (General Declarations) | 1 - 14 | ' Module : Report\_rptPipelineProblem ' Description: ' Procedures : Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| PageHeader0\_Format | 15 - 17 |  |
| Report\_Close | 19 - 34 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| Report\_Open | 36 - 58 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetText | 60 - 79 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| rptPipelineAction | (General Declarations) | 1 - 14 | ' Module : Report\_rptPipelineAction ' Description: ' Procedures : cmdDisplayGraph\_Click() ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail0\_Format | 15 - 21 |  |
| Report\_Close | 23 - 42 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- ' : June 26, 2009 mahmed ' enable on Application close button on close. '- |
| Report\_Open | 44 - 69 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetText | 71 - 97 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| rptPipelineProblem | (General Declarations) | 1 - 13 | ' Module : Report\_rptPipelineProblem ' Description: ' Procedures : Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Report\_Close | 14 - 32 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- ' : June 26, 2009 mahmed ' enable on Application close button on close. '- |
| Report\_Open | 34 - 58 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetText | 60 - 79 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| rptProcurementTable | (General Declarations) | 1 - 36 | '+ ' Module : Report\_rptProcurementTable ' Description: ' Procedures : Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() ' setYears() |
| Report\_Close | 37 - 59 |  |
| Report\_Open | 68 - 107 |  |
| SetText | 116 - 136 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetYears | 145 - 170 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| rptShipmentCostByFunding | (General Declarations) | 1 - 20 | '+ ' Module : Report\_rptShipmentCostBySupplier ' Description: ' Procedures : Detail\_Format(Cancel As Integer, FormatCount As Integer) ' PageFooter2\_Format(Cancel As Integer, FormatCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail\_Format | 29 - 50 | '+ ' Procedure : Detail1\_Format ' Comments : set flags for freight and cost ' Parameters: Cancel ' FormatCount - ' Modified : 01 Jul 1998 JSL ' 31 Jul 2003 LBlanken '- |
| PageFooter2\_Format | 61 - 74 |  |
| Report\_Close | 84 - 103 |  |
| Report\_Open | 113 - 144 |  |
| SetText | 154 - 169 |  |
| rptShipmentCostByFundingPortrait | (General Declarations) | 1 - 20 | '+ ' Module : Report\_rptShipmentCostBySupplier ' Description: ' Procedures : Detail\_Format(Cancel As Integer, FormatCount As Integer) ' PageFooter2\_Format(Cancel As Integer, FormatCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail\_Format | 29 - 50 | '+ ' Procedure : Detail1\_Format ' Comments : set flags for freight and cost ' Parameters: Cancel ' FormatCount - ' Modified : 01 Jul 1998 JSL ' 31 Jul 2003 LBlanken '- |
| PageFooter2\_Format | 61 - 74 |  |
| Report\_Close | 84 - 102 |  |
| Report\_Open | 112 - 144 |  |
| SetText | 154 - 169 |  |
| rptShipmentCostByProduct | (General Declarations) | 1 - 19 | '+ ' Module : Report\_rptShipmentCostByProduct ' Description: ' Procedures : Detail1\_Format(Cancel As Integer, FormatCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail1\_Format | 28 - 49 | '+ ' Procedure : Detail1\_Format ' Comments : set flags for freight and cost ' Parameters: Cancel ' FormatCount - ' Modified : 01 Jul 1998 JSL ' 31 Jul 2003 LBlanken '- |
| PageFooter2\_Format | 60 - 73 |  |
| Report\_Close | 86 - 104 |  |
| Report\_Open | 114 - 149 |  |
| SetText | 159 - 174 |  |
| rptShipmentCostByProductPortrait | (General Declarations) | 1 - 19 | '+ ' Module : Report\_rptShipmentCostByProduct ' Description: ' Procedures : Detail1\_Format(Cancel As Integer, FormatCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail1\_Format | 28 - 49 | '+ ' Procedure : Detail1\_Format ' Comments : set flags for freight and cost ' Parameters: Cancel ' FormatCount - ' Modified : 01 Jul 1998 JSL ' 31 Jul 2003 LBlanken '- |
| PageFooter2\_Format | 60 - 73 |  |
| Report\_Close | 86 - 105 |  |
| Report\_Open | 115 - 151 |  |
| SetText | 161 - 176 |  |
| rptShipmentCostBySupplier | (General Declarations) | 1 - 20 | '+ ' Module : Report\_rptShipmentCostBySupplier ' Description: ' Procedures : Detail\_Format(Cancel As Integer, FormatCount As Integer) ' PageFooter2\_Format(Cancel As Integer, FormatCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail\_Format | 29 - 50 | '+ ' Procedure : Detail1\_Format ' Comments : set flags for freight and cost ' Parameters: Cancel ' FormatCount - ' Modified : 01 Jul 1998 JSL ' 31 Jul 2003 LBlanken '- |
| PageFooter2\_Format | 61 - 74 |  |
| Report\_Close | 87 - 105 |  |
| Report\_Open | 115 - 149 |  |
| SetText | 159 - 174 |  |
| rptShipmentCostBySupplierPortrait | (General Declarations) | 1 - 20 | '+ ' Module : Report\_rptShipmentCostBySupplier ' Description: ' Procedures : Detail\_Format(Cancel As Integer, FormatCount As Integer) ' PageFooter2\_Format(Cancel As Integer, FormatCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail\_Format | 29 - 50 | '+ ' Procedure : Detail1\_Format ' Comments : set flags for freight and cost ' Parameters: Cancel ' FormatCount - ' Modified : 01 Jul 1998 JSL ' 31 Jul 2003 LBlanken '- |
| PageFooter2\_Format | 61 - 74 |  |
| Report\_Close | 87 - 108 |  |
| Report\_Open | 118 - 152 |  |
| SetText | 162 - 177 |  |
| rptShipmentOrders | (General Declarations) | 1 - 21 | '+ ' Module : Report\_rptShipmentOrders ' Description: ' Procedures : Detail\_Format(Cancel As Integer, FormatCount As Integer) ' GroupHeader1\_Format(Cancel As Integer, FormatCount As Integer) ' PageFooter2\_Format(Cancel As Integer, FormatCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail\_Format | 30 - 51 | '+ ' Procedure : Detail\_Format ' Comments : set flags for freight and cost ' Parameters: Cancel ' FormatCount - ' Modified : 01 Jul 1998 JSL ' 31 Jul 2003 LBlanken '- |
| GroupHeader1\_Format | 61 - 98 |  |
| PageFooter2\_Format | 109 - 122 |  |
| Report\_Close | 135 - 153 |  |
| Report\_Open | 163 - 201 |  |
| SetText | 211 - 226 |  |
| rptStatus\_mon | (General Declarations) | 1 - 22 | '+ ' Module : Report\_rptStatus\_mon ' Description: ' Procedures : Detail0\_Format(Cancel As Integer, FormatCount As Integer) ' getAdjustAmounts() ' getShipRecords() ' PageFooter1\_Format(Cancel As Integer, FormatCount As Integer) ' PageHeader0\_Format(Cancel As Integer, FormatCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail0\_Format | 31 - 44 | '+ ' Procedure : Detail0\_Format ' Comments : get adjustment amounts ' Parameters: Cancel ' FormatCount - ' Modified : 01 Jul 1998 JSL ' 11 Jul 2003 LBlanken '- |
| getAdjustAmounts | 54 - 91 |  |
| getShipRecords | 101 - 157 |  |
| PageFooter1\_Format | 168 - 182 |  |
| PageHeader0\_Format | 194 - 221 |  |
| Report\_Close | 233 - 251 |  |
| Report\_Open | 261 - 290 |  |
| SetText | 300 - 314 |  |
| rptStatus\_Qtr | (General Declarations) | 1 - 19 | '+ ' Module : Report\_rptStatus\_Qtr ' Description: ' Procedures : PageFooter1\_Format(Cancel As Integer, FormatCount As Integer) ' PageHeader0\_Format(Cancel As Integer, FormatCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| PageFooter1\_Format | 28 - 42 | '+ ' Procedure : PageFooter1\_Format ' Comments : show footer based on flag ' Parameters: Cancel ' FormatCount - ' Modified : 01 Jul 1998 JSL ' 11 Jul 2003 LBlanken '- |
| PageHeader0\_Format | 52 - 78 |  |
| Report\_Close | 90 - 109 |  |
| Report\_Open | 119 - 148 |  |
| SetText | 158 - 173 |  |
| rptStatusAllProducts\_mon | (General Declarations) | 1 - 22 | '+ ' Module : Report\_rptStatus\_mon ' Description: ' Procedures : Detail0\_Format(Cancel As Integer, FormatCount As Integer) ' getAdjustAmounts() ' getShipRecords() ' PageFooter1\_Format(Cancel As Integer, FormatCount As Integer) ' PageHeader0\_Format(Cancel As Integer, FormatCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail0\_Format | 31 - 44 | '+ ' Procedure : Detail0\_Format ' Comments : get adjustment amounts ' Parameters: Cancel ' FormatCount - ' Modified : 01 Jul 1998 JSL ' 11 Jul 2003 LBlanken '- |
| getAdjustAmounts | 54 - 91 |  |
| getShipRecords | 101 - 157 |  |
| GroupHeader1\_Format | 160 - 162 |  |
| PageFooter1\_Format | 172 - 186 |  |
| PageHeader0\_Format | 196 - 228 |  |
| Report\_Close | 240 - 258 |  |
| Report\_Open | 268 - 297 |  |
| SetText | 307 - 321 |  |
| rptStatusMatrix | (General Declarations) | 1 - 14 | ' Module : Report\_rptAnnualCosts ' Description: ' Procedures : GroupHeader5\_Print(Cancel As Integer, PrintCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail1\_Format | 15 - 28 |  |
| Detail1\_Print | 109 - 122 |  |
| FormatCell | 227 - 273 | ' Comments : ' Anytime the MOS calculation showed stock levels of a product above max, the number in that cell would appear in italics. If the MOS calculation showed stocks below min, the cell holding the MOS for that particular product would be highlighted. When the product was stocked out, the cell would be highlighted more strongly. ' Parameters : ' Returns : ' Created : ' -------------------------------------------------------- |
| GroupHeader0\_Print | 30 - 68 | ' Comments : ' Parameters : Cancel ' PrintCount ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| GroupHeader5\_Print | 70 - 107 | ' Comments : ' Parameters : Cancel ' PrintCount ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| PageHeader0\_Format | 124 - 131 |  |
| Report\_Close | 133 - 162 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- ' : June 26, 2009 mahmed ' enable on Application close button on close. '- |
| Report\_Open | 164 - 190 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetText | 192 - 222 | ' Comments : ' Parameters : ' Returns : ' Created : 06/16/98 JSL ' Modified : ' ' July 21, 2009 mahmed ' report subtitle now created from user-defined range ' -------------------------------------------------------- |
| rptStatusMatrixMethod | (General Declarations) | 1 - 11 | ' Module : Report\_rptStatusMatrixMethod ' Description: ' Procedures : |
| Detail1\_Format | 12 - 42 |  |
| FormatCell | 248 - 288 | ' Comments : ' Anytime the MOS calculation showed stock levels of a product above max, the number in that cell would appear in italics. If the MOS calculation showed stocks below min, the cell holding the MOS for that particular product would be highlighted. When the product was stocked out, the cell would be highlighted more strongly. ' Parameters : ' Returns : ' Created : ' -------------------------------------------------------- |
| GroupHeader0\_Print | 44 - 82 | ' Comments : ' Parameters : Cancel ' PrintCount ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| GroupHeader5\_Print | 84 - 121 | ' Comments : ' Parameters : Cancel ' PrintCount ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| PageHeader0\_Format | 123 - 131 |  |
| Report\_Close | 133 - 161 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- ' : June 26, 2009 mahmed ' enable on Application close button on close. '- |
| Report\_NoData | 163 - 183 | ' Comments : ' Parameters: Cancel - ' Modified : ' ' -------------------------------------------------- |
| Report\_Open | 185 - 211 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| SetText | 213 - 243 | ' Comments : ' Parameters : ' Returns : ' Created : 06/16/98 JSL ' Modified : ' ' July 21, 2009 mahmed ' report subtitle now created from user-defined range ' -------------------------------------------------------- |
| rptStatusMatrixProduct | (General Declarations) | 1 - 10 | ' Module : Report\_rptStatusMatrixProduct ' Description: ' Procedures : |
| Detail1\_Format | 11 - 41 |  |
| FormatCell | 251 - 294 | ' Comments : ' Anytime the MOS calculation showed stock levels of a product above max, ' the number in that cell would appear in italics. If the MOS calculation showed ' stocks below min, the cell holding the MOS for that particular product would be highlighted. ' When the product was stocked out, the cell would be highlighted more strongly. ' Parameters : ' Returns : ' Created : 16 Sep 2009 mahmed ' -------------------------------------------------------- |
| GroupHeader0\_Print | 43 - 81 | ' Comments : ' Parameters : Cancel ' PrintCount ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| GroupHeader5\_Print | 83 - 120 | ' Comments : ' Parameters : Cancel ' PrintCount ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- |
| PageHeader0\_Format | 122 - 130 |  |
| Report\_Close | 132 - 160 | ' Comments : ' Parameters : ' Returns : ' Created : ' Modified : 01 Jul 1998 JSL ' ' -------------------------------------------------------- ' : June 26, 2009 mahmed ' enable on Application close button on close. '- |
| Report\_NoData | 162 - 182 | ' Comments : ' Parameters: Cancel - ' Modified : ' ' -------------------------------------------------- |
| Report\_Open | 186 - 213 | ' Comments : ' Parameters : Cancel ' Returns : ' Created : ' Modified : ' ' -------------------------------------------------------- |
| SetText | 215 - 245 | ' Comments : ' Parameters : ' Returns : ' Created : 06/16/98 JSL ' Modified : ' ' July 21, 2009 mahmed ' report subtitle now created from user-defined range ' -------------------------------------------------------- |
| rptStatusMethod\_mon | (General Declarations) | 1 - 22 | '+ ' Module : Report\_rptStatusMethod\_mon ' Description: ' Procedures : Detail0\_Format(Cancel As Integer, FormatCount As Integer) ' getAdjustAmounts() ' getShipRecords() ' PageFooter1\_Format(Cancel As Integer, FormatCount As Integer) ' PageHeader0\_Format(Cancel As Integer, FormatCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail0\_Format | 31 - 44 | '+ ' Procedure : Detail0\_Format ' Comments : get adjustment amounts ' Parameters: Cancel ' FormatCount - ' Modified : 01 Jul 1998 JSL ' 11 Jul 2003 LBlanken '- |
| getAdjustAmounts | 54 - 91 |  |
| getShipRecords | 101 - 157 |  |
| PageFooter1\_Format | 168 - 182 |  |
| PageHeader0\_Format | 194 - 226 |  |
| Report\_Close | 238 - 258 |  |
| Report\_Open | 268 - 297 |  |
| SetText | 307 - 321 |  |
| rptStatusMethod\_QTR | (General Declarations) | 1 - 22 | '+ ' Module : Report\_rptStatusMethod\_qtr ' Description: ' Procedures : Detail0\_Format(Cancel As Integer, FormatCount As Integer) ' getAdjustAmounts() ' getShipRecords() ' PageFooter1\_Format(Cancel As Integer, FormatCount As Integer) ' PageHeader0\_Format(Cancel As Integer, FormatCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail0\_Format | 31 - 44 | '+ ' Procedure : Detail0\_Format ' Comments : get adjustment amounts ' Parameters: Cancel ' FormatCount - ' Modified : 01 Jul 1998 JSL ' 11 Jul 2003 LBlanken '- |
| getAdjustAmounts | 54 - 91 |  |
| getShipRecords | 101 - 157 |  |
| PageFooter1\_Format | 168 - 182 |  |
| PageHeader0\_Format | 194 - 227 |  |
| Report\_Close | 238 - 256 |  |
| Report\_Open | 266 - 295 |  |
| SetText | 305 - 319 |  |
| rptSupplyPlan | (General Declarations) | 1 - 22 | '+ ' Module : Report\_rptStatus\_mon ' Description: ' Procedures : Detail0\_Format(Cancel As Integer, FormatCount As Integer) ' getAdjustAmounts() ' getShipRecords() ' PageFooter1\_Format(Cancel As Integer, FormatCount As Integer) ' PageHeader0\_Format(Cancel As Integer, FormatCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail0\_Format | 31 - 44 | '+ ' Procedure : Detail0\_Format ' Comments : get adjustment amounts ' Parameters: Cancel ' FormatCount - ' Modified : 01 Jul 1998 JSL ' 11 Jul 2003 LBlanken '- |
| getAdjustAmounts | 54 - 68 |  |
| getShipRecords | 78 - 134 |  |
| PageFooter1\_Format | 145 - 158 |  |
| PageHeader0\_Format | 170 - 196 |  |
| Report\_Close | 205 - 223 |  |
| Report\_NoData | 226 - 246 | ' Comments : ' Parameters: Cancel - ' Modified : ' ' -------------------------------------------------- |
| Report\_Open | 255 - 284 |  |
| SetText | 294 - 308 |  |
| rptSupplyPlanBySupplier | (General Declarations) | 1 - 22 | '+ ' Module : Report\_rptStatus\_mon ' Description: ' Procedures : Detail0\_Format(Cancel As Integer, FormatCount As Integer) ' getAdjustAmounts() ' getShipRecords() ' PageFooter1\_Format(Cancel As Integer, FormatCount As Integer) ' PageHeader0\_Format(Cancel As Integer, FormatCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| Detail0\_Format | 31 - 44 | '+ ' Procedure : Detail0\_Format ' Comments : get adjustment amounts ' Parameters: Cancel ' FormatCount - ' Modified : 01 Jul 1998 JSL ' 11 Jul 2003 LBlanken '- |
| getAdjustAmounts | 54 - 68 |  |
| getShipRecords | 78 - 134 |  |
| PageFooter1\_Format | 145 - 158 |  |
| PageHeader0\_Format | 172 - 196 |  |
| Report\_Close | 205 - 222 |  |
| Report\_Open | 232 - 260 |  |
| SetText | 270 - 284 |  |
| rsubReportComments | (General Declarations) | 1 - 12 | ' Module : Report\_rsubReportComments ' Description: ' Procedures : SetText() ' Report\_Open(Cancel As Integer) |
| Report\_Open | 24 - 44 | ' Comments : ' Parameters: Cancel - ' Modified : ' ' -------------------------------------------------- |
| SetText | 13 - 22 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| rsubReportCommentsLS | (General Declarations) | 1 - 12 | ' Module : Report\_rsubReportComments ' Description: ' Procedures : SetText() ' Report\_Open(Cancel As Integer) |
| Report\_Open | 24 - 33 | ' Comments : ' Parameters: Cancel - ' Modified : ' ' -------------------------------------------------- |
| SetText | 13 - 22 | ' Comments : ' Parameters: - ' Modified : ' ' -------------------------------------------------- |
| srptImport | (General Declarations) | 1 - 2 |  |
| Report\_Open | 3 - 5 |  |
| SetText | 13 - 28 | '+ ' Procedure : SetText ' Comments : set text on report ' Parameters: - ' Modified : 01 Jul 1998 JSL ' 31 Jul 2003 LBlanken '- |
| srptImportInfo | (General Declarations) | 1 - 15 | '+ ' Module : Report\_rptiMPORT ' Description: ' Procedures : GroupHeader1\_Format(Cancel As Integer, FormatCount As Integer) ' Report\_Close() ' Report\_Open(Cancel As Integer) ' SetText() |
| GroupHeader1\_Format | 23 - 58 | '+ ' Procedure : GroupHeader1\_Format ' Comments : set captions for status' ' Parameters: Cancel ' FormatCount - ' Modified : 31 Jul 2003 LBlanken '- |
| GroupHeader2\_Format | 60 - 64 |  |
| Report\_Open | 72 - 92 | '+ ' Procedure : Report\_Open ' Comments : set captions and text of report and maximize ' Parameters: Cancel - ' Modified : 01 Jul 1998 JSL ' 31 Jul 2003 LBlanken '- |
| SetText | 102 - 117 |  |
| srptImportShipments | (General Declarations) | 1 - 4 |  |
| Report\_Close | 5 - 15 |  |
| Report\_Open | 17 - 20 |  |
| SetText | 28 - 43 | '+ ' Procedure : SetText ' Comments : set text on report ' Parameters: - ' Modified : 01 Jul 1998 JSL ' 31 Jul 2003 LBlanken '- |
| srptImportShipmentsImported | (General Declarations) | 1 - 10 |  |
| Detail\_Format | 11 - 32 |  |
| Report\_Open | 34 - 36 |  |
| ReportFooter\_Format | 62 - 64 |  |
| SetText | 44 - 60 | '+ ' Procedure : SetText ' Comments : set text on report ' Parameters: - ' Modified : 01 Jul 1998 JSL ' 31 Jul 2003 LBlanken '- |
| srptImportShipmentsSkipped | (General Declarations) | 1 - 10 |  |
| Detail\_Format | 11 - 32 |  |
| Report\_Open | 34 - 36 |  |
| ReportFooter\_Format | 61 - 63 |  |
| SetText | 44 - 59 | '+ ' Procedure : SetText ' Comments : set text on report ' Parameters: - ' Modified : 01 Jul 1998 JSL ' 31 Jul 2003 LBlanken '- |

Modules

Table - List of Modules

| ModuleName | Procedure | Lines | Comments |
| --- | --- | --- | --- |
| App Constants and Variables | (General Declarations) | 1 - 72 | '+ ' Module : App Constants and Variables ' Description: ' Procedures : ' Modified : ' 05/22/03 LBlanken Cleaned with Total Visual CodeTools ' 12 May 2004 LKB '- |
| basAccessConstants | (General Declarations) | 1 - 220 | '+ ' Module : basAccessConstants ' Description: ' Procedures : ' Modified : ' 05/22/03 LBlanken Cleaned with Total Visual CodeTools ' 12 May 2004 LKB '- |
| basAPIFunctions | (General Declarations) | 1 - 253 | '+ ' Module : basAPIFunctions ' Description: ' Procedures : ConvertTSB2Win(TSB\_Struct As TSBAPI\_OPENFILE, Win\_Struct As TSBAPI\_WINOPENFILENAME) ' ConvertWin2TSB(Win\_Struct As TSBAPI\_WINOPENFILENAME, TSB\_Struct As TSBAPI\_OPENFILE) ' getFileDialog\_tsb(strInitialDir As String, strTitle As String, strFilter As String, strFile As String) ' GetOpenFile\_TSB(strInitialDir As String, strTitle As String, strFilter As String, strFile As String) ' GetUserName\_TSB() ' RemoveNulls\_TSB(strIN As String) ' SZToString\_TSB(strIN As String) ' Modified : ' 28 Jul 2003 LBlanken '- |
| AppCloseEnabled | 552 - 566 |  |
| ConvertTSB2Win | 261 - 302 | '+ ' Procedure : ConvertTSB2Win ' Comments : Converts the passed TSBAPI structure to a Windows structure ' Parameters: TSB\_Struct - record of type TSBAPI\_OPENFILE ' Win\_Struct - record of type TSBAPI\_WINOPENFILENAME ' Modified : 28 Jul 2003 LBlanken '- |
| ConvertWin2TSB | 312 - 327 |  |
| getFileDialog\_tsb | 342 - 381 |  |
| GetOpenFile\_TSB | 394 - 441 |  |
| GetUserName\_TSB | 451 - 478 |  |
| RemoveNulls\_TSB | 488 - 510 |  |
| SZToString\_TSB | 521 - 542 |  |
| basAppConstants | (General Declarations) | 1 - 131 | '+ ' Module : App Constants ' Description: Constants used in the applications ' Procedures : ' Modified : 04 Sep 2003 LKB ' 1 Sep 2009 mahmed - add constants for shipment import function '- |
| basCheckAccessVersion | (General Declarations) | 1 - 2 |  |
| CheckAccessVersion | 3 - 25 |  |
| basCloneConsumption | (General Declarations) | 1 - 15 | '+ ' Module : basCloneConsumption ' Description : ' Procedures : CheckCloneDataSource() ' CloneConsumption(strProductID As String) |
| CheckCloneDataSource | 22 - 48 | '+ ' Procedure : CheckCloneDataSource ' Comments : Validates the Clone Data Source exists ' Parameters : - ' Modified : 18 Aug 2009 jsl '- |
| CloneConsumption | 58 - 99 |  |
| basCompactDB | (General Declarations) | 1 - 19 | '+ ' Module : basCompactDB ' Description: ' Procedures : CompactBackend() ' DoBackendCompact() ' GetBackendSize() ' GetBackendStartSize() ' GetDatabaseSize() ' SetBackendStartSize() ' SetCompactOnExit() |
| CompactBackend | 27 - 75 | '+ ' Procedure : CompactBackend ' Comments : Determine if the Backend database has grown sufficiently to Compact Now. ' Parameters: - ' Created : 09-Jan-02 PM JSL ' Modified : 12 May 2004 LKB '- |
| DoBackendCompact | 85 - 98 |  |
| GetBackendSize | 108 - 130 |  |
| GetBackendStartSize | 140 - 154 |  |
| GetDatabaseSize | 164 - 184 |  |
| SetBackendStartSize | 194 - 214 |  |
| SetCompactOnExit | 225 - 255 |  |
| basCPTData | (General Declarations) | 1 - 18 | '+ ' Module : basCPTData ' Description: ' Procedures : GenCountryFile() ' GenCPTData() ' GenMethods() ' GenProducts() ' GenShipments(strYear As String) ' GenSupplierData() |
| GenCountryFile | 27 - 80 | '+ ' Procedure : GenCountryFile ' Comments : To generate the Country File for the CPT Data. ' Parameters: - ' Returns : Integer - ' Created : 05/26/98 JSL ' Modified : 01 Jul 1998 JSL ' - |
| GenCPTData | 91 - 372 |  |
| GenMethods | 383 - 436 |  |
| GenProducts | 447 - 504 |  |
| GenShipments | 516 - 604 |  |
| GenSupplierData | 614 - 666 |  |
| basExcelAdmin | (General Declarations) | 1 - 70 |  |
| ahtAddFilterItem | 222 - 233 | ' Tack a new chunk onto the file filter. ' That is, take the old value, stick onto it the description, ' (like "Databases"), a null character, the skeleton ' (like "\*.mdb;\*.mda") and a final null character. |
| ahtCommonFileOpenSave | 125 - 220 | ' This is the entry point you'll use to call the common ' file open/save dialog. The parameters are listed ' below, and all are optional. ' ' In: ' Flags: one or more of the ahtOFN\_\* constants, OR'd together. ' InitialDir: the directory in which to first look ' Filter: a set of file filters, set up by calling ' AddFilterItem. See examples. ' FilterIndex: 1-based integer indicating which filter ' set to use, by default (1 if unspecified) ' DefaultExt: Extension to use if the user doesn't enter one. ' Only useful on file saves. ' FileName: Default value for the file name text box. ' DialogTitle: Title for the dialog. ' hWnd: parent window handle ' OpenFile: Boolean(True=Open File/False=Save As) ' Out: ' Return Value: Either Null or the selected filename |
| GetOpenFile | 88 - 123 | ' Here's an example that gets an Access database name. |
| TestIt | 71 - 86 | ' ' Dim Strfilter As String ' Dim lngFlags As Long ' Strfilter = ahtAddFilterItem(Strfilter, "Access Files (\*.mda, \*.mdb)", \_ ' "\*.MDA;\*.MDB") ' Strfilter = ahtAddFilterItem(Strfilter, "dBASE Files (\*.dbf)", "\*.DBF") ' Strfilter = ahtAddFilterItem(Strfilter, "Text Files (\*.txt)", "\*.TXT") ' Strfilter = ahtAddFilterItem(Strfilter, "All Files (\*.\*)", "\*.\*") ' MsgBox "You selected: " & ahtCommonFileOpenSave(InitialDir:="C:\", \_ ' filter:=Strfilter, FilterIndex:=3, Flags:=lngFlags, \_ ' DialogTitle:="Hello! Open Me!") ' ' Since you passed in a variable for lngFlags, ' ' the function places the output flags value in the variable. ' Debug.Print Hex(lngFlags) |
| TrimNull | 235 - 243 |  |
| basExportReports | (General Declarations) | 1 - 13 |  |
| DetectExcel | 20 - 43 | '+ ' Procedure : DetectExcel ' Comments : ' Parameters : - ' Modified : 08 Dec 2010 JSL '- |
| ExportReport | 53 - 426 |  |
| GetExcel | 435 - 476 |  |
| TableExists | 486 - 495 |  |
| basFileFunctions | (General Declarations) | 1 - 19 | '+ ' Module : basFileFunctions ' Description: ' Procedures : DirExists\_TSB(strDir As String) ' FileExists\_TSB(strDest As String) ' GetCodeDBNamePath\_TSB() ' GetNamePart\_TSB(strIN As String) ' GetPathPart\_TSB(strPath As String) ' GetPmppPath() ' ParsePath\_TSB(strIN As String, strDrive As String, strPath As String, strFileName As String, strExtension As String) |
| DirExists\_TSB | 27 - 41 | '+ ' Procedure : DirExists\_TSB ' Comments : determines if the named directory exists ' Parameters: strDir - directory to check ' Returns : True - directory exists, False otherwise ' Modified : 12 May 2004 LKB ' - |
| FileExists\_TSB | 52 - 61 |  |
| GetCodeDBNamePath\_TSB | 71 - 89 |  |
| GetNamePart\_TSB | 99 - 124 |  |
| GetPathPart\_TSB | 134 - 156 |  |
| GetPmppPath | 167 - 192 |  |
| ParsePath\_TSB | 206 - 264 |  |
| basFormsUtils | (General Declarations) | 1 - 39 | '+ ' Module : basFormsUtils ' Description: Utility function run from forms ' Procedures : ApplyFilter(frmIn As Form) ' BuildTmpSort(lngChild As Long ' CountryFormHead() As String ' DisableFrmCtls(frmIn As Form, intSection As Integer, fEnable As Boolean) ' FindCategoryChild(varParent As Variant) ' GenerateCodes(strName As String, strForm As String) ' getText(strTextID As String) ' IsLoaded(strFormName As String) ' IsLoadedRep(strRepName As String) ' listboxReset(lstBox As ListBox) ' MakeUSDATE(varDateIn As Variant) ' OpenCostReport(strByType As String) ' OpenDB(strDB As String) ' OpenPipeLineSummary() ' ProgFormHead() ' SaveForm(Frmin As Form) ' SetCboSelect2(strParent As String, ctl As Control) ' ShowHideSubform(formIn As Form, strSubForm As String) ' xfertoForm(strToForm As String, strFromForm As String, intKillMe ' As Integer) |
| AddShipment | 47 - 122 | '+ ' Procedure : AddShipment ' Comments : Sets date to end of month and cycles through items in listbox ' : creating a shipment record for each ' Parameters: - ' Modified : 27 Jun 2003 LBlanken '- |
| AdjustForCase | 131 - 217 |  |
| ApplyFilter | 227 - 260 |  |
| ApplyReportFilter | 270 - 303 |  |
| AskSaveFormPopup | 1764 - 1782 |  |
| BuildFromYears | 315 - 360 |  |
| BuildThroughYears | 373 - 415 |  |
| BuildTmpSort | 425 - 455 |  |
| BuildYears | 467 - 494 |  |
| CheckInterpolateForm | 2199 - 2236 |  |
| checkMonthSelected | 2501 - 2516 |  |
| CheckValue | 506 - 527 |  |
| CountryFormHead | 538 - 560 |  |
| CreateTablePointer | 1984 - 2012 | ' Comments : Create a pointer on a attached table to open ' : a recordset as a table type ' Parameters : strTable - the table to create the link on ' Returns : Recordset - ' Created : ' Modified : ' ' -------------------------------------------------------- |
| DisableFrmCtls | 571 - 607 |  |
| DomainLookup\_TSB | 1608 - 1652 | ' Comments : Returns a field value in a specified set of records ' Parameters: strDatabase - path and name of database to look in or "" (blank string) for the current database ' strField - name of the field to return ' strDomain - name of the table or query to search in ' strCriteria - string expression specifying the WHERE clause of the query or blank for no constraints (all records) ' Returns : value of the specified field as a variant or NULL if no records matching strCriteria are found ' |
| ErrorWithData | 617 - 632 |  |
| FindCategoryChild | 642 - 688 |  |
| GenerateCodes | 700 - 784 |  |
| getAdjustAmounts | 2410 - 2437 |  |
| getAdjustAmountsMethod | 2440 - 2467 |  |
| getAMCFlag | 2476 - 2489 |  |
| getDefaultPeriod | 2273 - 2295 | ' Comments : ' Parameters: ' Returns : Integer - ' Created : 08/19/09 15:36 mahmed ' Modified : ' ' -------------------------------------------------- |
| getFundingSourceList | 2367 - 2407 |  |
| getFundingSourceName | 2344 - 2355 |  |
| GetProductName | 2251 - 2272 |  |
| getText | 796 - 839 |  |
| IsLoaded | 850 - 875 |  |
| IsLoadedRep | 886 - 908 |  |
| listboxReset | 921 - 970 |  |
| MakeUSDATE | 982 - 1002 |  |
| MakeYears | 1012 - 1062 |  |
| OpenCostReport | 1072 - 1099 |  |
| OpenDB | 1109 - 1141 |  |
| OpenPipeLineSummary | 1152 - 1261 |  |
| ProgFormHead | 1272 - 1286 |  |
| RequerySubform | 1296 - 1376 |  |
| SanitizeFileNameOrPath | 1694 - 1762 | ' Comments : Return name with invalid characters replaced by "\_" ' All characters greater than ASCII 31 to be used except for the following: "\*/:<>?\| ' The name may not be only dots ' Parameters : Candidate file name ' Created : Jan 18, 2007 - lblanken - Bug: 13663 ' -------------------------------------------------------- |
| SaveForm | 1387 - 1416 |  |
| SetCboSelect2 | 1427 - 1456 |  |
| SetDateRanges | 1799 - 1979 |  |
| SetDropDownWidth | 2298 - 2335 | '+ ' Procedure : SetDropDownWidth ' Comments : sets list width property of list box to longest entry in the list ' Parameters: - ' created : 26 July 2009 MAhmed '- |
| SetInterpolateForm | 2022 - 2185 |  |
| ShowHideMain | 1499 - 1515 | '+ ' Procedure : ShowHideMain ' Comments : Shows/Hides main based on global mode ' Parameters: FormIn - current form ' strSubForm - source of subform ' Modified : 27 Jun 2003 LBlanken '- |
| ShowHideSubform | 1466 - 1491 |  |
| UpdateCommandButtons | 1654 - 1693 |  |
| UpdateControls | 1526 - 1557 |  |
| xfertoForm | 1571 - 1606 |  |
| basGetExcel | (General Declarations) | 1 - 2 |  |
| getexcelpath | 3 - 62 |  |
| basGetStatus | (General Declarations) | 1 - 17 | '+ ' Module : basgetstatus ' Description: Set the form and report controls captions, tooltips, styles ' and statusbar text also gets the status text and report text ' Procedures : SetFormStatusText(frmIn As Form, Optional bytIndex As Byte, ' Optional bytType As Byte) ' SetReportCaptions(rptIn As Report, Optional bytIndex As Byte) ' GetStatusText(strForm As String, strCTL As String, bytIndex ' As Byte) ' Modified : 22 May 2003 LBlanken Cleaned with Total Visual CodeTools ' 16 Jun 2003 LBlanken '- |
| GetStatusText | 29 - 80 | '+ ' Procedure : GetStatusText ' Comments : returns the text for the statusbar for the control as found ' in table ' Parameters: strForm - form ' strCTL - control on form ' bytIndex - ' Returns : String - ' Created : 02 Aug 2000 Jeff Leiner ' Modified : 16 Jun 2003 LBlanken ' - |
| SetFormStatusText | 94 - 214 |  |
| SetReportCaptions | 227 - 315 |  |
| basImport | (General Declarations) | 1 - 9 |  |
| ClearProcessed | 843 - 860 |  |
| ClearProcessedShipments | 1057 - 1074 |  |
| ClearUnMapped | 556 - 573 | '+ ' Procedure : ClearUnMapped ' Comments : Removes all unmapped data from the Import tables ' Parameters: - ' Modified : 17 Jul 2006 MAT '- |
| ImportForecastData | 575 - 730 |  |
| ImportProducts | 10 - 549 |  |
| ImportShipments | 910 - 1049 |  |
| Validate\_Period | 743 - 835 | '+ ' Procedure : Validate\_Period ' Comments : Returns true if at least part of time period entered is unique ' (accept edits). Returns false if at least part of time period ' not unique (reject edits). Unique means time period not duplicated ' by another record of the same type. Actual consumption can duplicate ' a forecast time period ' Parameters: - ' Returns : - ' Modified : 01 Jul 1998 JSL ' 16 Jun 2003 LBlanken ' - |
| ValidateFile | 862 - 908 |  |
| basImportNewvernshipments | (General Declarations) | 1 - 14 | '+ ' Module : basImportNewvernshipments ' Description: ' Procedures : GetNewvernShipmentFile() |
| GetNewvernShipmentFile | 23 - 161 | '+ ' Procedure : GetNewvernShipmentFile ' Comments : Find a newvern shipment file and import it into the .DB ' Parameters: - ' Returns : Boolean - ' Created : 06-Mar-00 Jeff Leiner ' Modified : 12 May 2004 LKB ' - |
| basMiscTools | (General Declarations) | 1 - 24 | '+ ' Module : Misc\_Tools ' Description: ' Procedures : BuildInString(lst As ListBox, fIsString As Boolean) ' CountOccurrences\_TSB(strIn As String, strFind As String) ' FillString\_TSB(strChar As String, intCount As Integer) ' MaxOfThree\_TSB(varValue1 As Variant, varValue2 As Variant, varValue3 As Variant) ' Movepointer(frmActive As Form, varDirection As Variant) ' NullToValue(varNull As Variant, strValue As String) ' Open\_Admin\_Form(strFormName As String) ' Open\_Form(strFormName As String) ' Open\_Supervisor\_Form(strFormName As String) ' Return\_To\_Form(strFormName As String) ' StripChars\_TSB(strIn As String, strStrip As String) |
| BuildExclSupplierNameString | 713 - 767 |  |
| BuildInString | 35 - 72 | '+ ' Procedure : BuildInString ' Comments : Build a Instring statement from the selected values ' : in a multiselect listbox ' Parameters: lst - The listbox to use ' fIsString - If the value is a string it must be in single quote ' Returns : String - ' Created : 05 Apr 2000 Jeff Leiner ' Modified : 20 Jun 2003 LBlanken ' - |
| ceiling | 658 - 673 |  |
| CountDelimitedWords\_TSB | 591 - 610 | ' Comments : returns the number of words in a delimited string ' Parameters: strIn - string to count words in ' chrDelimit - character that delimits words in strIn ' Returns : number of occurrences ' |
| CountOccurrences\_TSB | 83 - 122 |  |
| FillString\_TSB | 133 - 156 |  |
| GetDelimitedWord\_TSB | 613 - 656 | ' Comments : returns word intIndex in delimited string strIn ' Parameters : strIn - string to search ' intIndex - word position to find (value of 0 or greater than word count is undefined.) ' chrDelimit - character used as the delimter ' Returns : nth word ' |
| GetValue | 882 - 899 |  |
| IsCharLetter\_TSB | 512 - 531 | ' Comments : determines if the passed character is an letter (a-z, A-Z, à-ÿ, À-Þ) ' Parameters: varChar - Character as variant ' Returns : True - character is alpha, False - character is not alpha ' |
| IsCharNumeric\_TSB | 534 - 545 | ' Comments : determines if the passed character is numeric ' Parameters: varChar - character to check ' Out : True - character is numeric, False - character is not numeric ' |
| MakeDataSetMatrix | 776 - 845 |  |
| MaxOfThree\_TSB | 168 - 220 |  |
| Movepointer | 232 - 314 |  |
| ntz | 846 - 871 | ' Comments : Returns a 0 if the value is null or the same value if not null ' Useful in queries and calculations where there is missing data ' Parameters : varValue - value to test ' Returns : 0 if null, original value if not null ' Created : ' Modified : ' ' -------------------------------------------------------- |
| NullToValue | 326 - 340 |  |
| Open\_Admin\_Form | 351 - 367 |  |
| Open\_Form | 378 - 390 |  |
| Open\_Supervisor\_Form | 401 - 419 |  |
| ReplaceString\_TSB | 484 - 508 | ' Comments : replaces a substring in a string with another ' Parameters : strTextIn - string to work on ' strFind - string to find ' strReplace - string to replace with ' fCaseSensitive - True for case sensitive search, False for case-insensitive search ' Returns : modified string ' |
| Return\_To\_Form | 430 - 442 |  |
| RoundtoNextNumber | 674 - 701 | ' Comments : Rounds a number to a specified number of decimal places (if any ' : remainder than the value is rounded up). ' Parameters: dblValue - number to round ' intDecimals - number of decimal places to round to ' (positive for right of decimal, negative for left) ' Returns : Rounded number ' Created : ' Modified : ' ' -------------------------------------------------------- |
| StringToArray\_TSB | 570 - 589 | ' Comments : Converts a delimited string to an array of words ' Parameters: strIn - string to convert ' arrIn - array of strings (1-based) ' chrDelimit - character used to delimit words in strIn ' Returns : number of words ' |
| StripChars\_Special | 546 - 566 |  |
| StripChars\_TSB | 453 - 482 |  |
| basMsgBoxUnicode | (General Declarations) | 1 - 16 | '+ ' Module : basMsgBoxUnicode ' Description : Functions for producing a unicode compliant message box ' Procedures : MsgBoxU() ' Modified : 09 Nov 2005 JSL '- |
| MsgBoxU | 31 - 73 | '+ ' Procedure : MsgBoxU ' Comments : Unicode enabled Messagebox wrapper, accepting all arguments ' : although help doesn't work at this time. ' Parameters : Prompt - the Message to display ' : lngButtons - the msgbox options (of vbMsgboxStyle) ' : Title - The title of the msgbox. if null then application name ' : - is displayed. ' : HelpFile = the path to the help file (N/A) ' : Context - the help context ID (N/A) ' Returns : Long - the result of the message box ' Created : 09 Nov 2005 jleiner ' Modified : '- |
| basPDFOutput | (General Declarations) | 1 - 5 |  |
| ChoosePDFPrinter | 83 - 87 |  |
| GetPDFEngine | 122 - 146 |  |
| IsReportLoaded | 103 - 105 |  |
| printreportPDF | 6 - 79 |  |
| SetDefaultPrinter | 89 - 101 |  |
| SetOpen | 107 - 111 |  |
| SetPDFEngine | 147 - 174 |  |
| basPriceFunctions | (General Declarations) | 1 - 22 | '+ ' Module : basPriceFunctions ' Description: ' Procedures : EstOrderDate(lngShipID As Long) ' GetDefaultSupplier(strProductID) ' getFreightRate(lngShipID As Long) ' getheader(Col) ' GetUnitvalue(lngShipment As Long) ' Round(ByVal dblNumber As Double, ByVal intDecimals As Integer) ' xtabprint(lngView As Long) |
| EstOrderDate | 32 - 71 | '+ ' Procedure : EstOrderDate ' Comments : When a shipments Order Date is not specified caculate its ' its estimated Date and return the value ' Parameters: lngShipID - ' Returns : Variant - The Estimated Date ' Created : 06/10/98 JSL ' Modified : 12 May 2004 LKB ' - |
| GetDefaultSupplier | 82 - 104 |  |
| getFreightRate | 118 - 185 |  |
| getheader | 196 - 212 |  |
| GetUnitvalue | 222 - 294 |  |
| Round | 305 - 323 |  |
| xtabprint | 335 - 405 |  |
| xtabprintByFunding | 416 - 488 |  |
| xtabprintBySupplier | 498 - 570 |  |
| xtabprintStatusMethod | 657 - 725 |  |
| xtabprintStatusProduct | 579 - 646 | '+ ' Procedure : xtabprintStatusProduct ' Comments : to Dynamically create the data for a report based on a ' : Crosstab query ' Parameters: lngView - ' Returns : - ' Created : 11/08/2009 mahmed ' - |
| basReferences | (General Declarations) | 1 - 14 | '+ ' Module : basReferences ' Description: ' Procedures : ReferenceInfo() |
| ReferenceInfo | 22 - 50 | '+ ' Procedure : ReferenceInfo ' Comments : ' Parameters: - ' Returns : - ' Modified : 12 May 2004 LKB ' - |
| basRegistryExcel | (General Declarations) | 1 - 90 |  |
| fReturnRegKeyValue | 91 - 167 |  |
| basRegistrySettings | (General Declarations) | 1 - 111 | '+ ' Module : basRegistrySettings ' Description: ' Procedures : GetAllowCPTExports() ' GetNewInstall() ' RegGetKeyValue\_TSB(lngRootKey As Long, strKeyName As String, strValueName As String) ' RegSetKeyValue\_TSB(lngRootKey As Long, strKeyName As String, strValueName As String, varValue As Variant, lngValueType As Long) ' SetNewInstall(strValue As String) |
| GetAllowCPTExports | 119 - 141 | '+ ' Procedure : GetAllowCPTExports ' Comments : ' Parameters: - ' Returns : Boolean - ' Modified : 12 May 2004 LKB ' - |
| GetNewInstall | 151 - 173 |  |
| RegCreateNewKey\_TSB | 339 - 361 | '+ ' Procedure : RegCreateNewKey\_TSB ' Comments : Creates a new key ' Parameters: lngRootKey - root key value, must be one of the following constants ' regHKeyClassesRoot ' regHKeyCurrentUser ' regHKeyLocalMachine ' regHKeyUsers ' strKeyName - The name of the key to create ' Returns : True if successful, False otherwise ' Modified : 30 Jul 2003 JLeiner ' - |
| RegDeleteKey\_TSB | 374 - 390 | '+ ' Procedure : RegDeleteKey\_TSB ' Comments : Deletes the specified key from the system registry ' Parameters: lngRootKey - root key value, must be one of the following constants ' regHKeyClassesRoot ' regHKeyCurrentUser ' regHKeyLocalMachine ' regHKeyUsers ' strKeyName - The name of the key to delete ' Returns : True if successful, False otherwise ' Modified : 30 Jul 2003 JLeiner ' - |
| RegDeleteValue\_TSB | 406 - 428 |  |
| RegGetKeyValue\_TSB | 189 - 251 |  |
| RegSetKeyValue\_TSB | 269 - 301 | ' |
| SetNewInstall | 310 - 326 |  |
| basReportComments | (General Declarations) | 1 - 16 | '+ ' Module : basReportComments ' Description: ' Procedures : GetReportComments(intReportType As Integer, dtmStart As Date, dtmEnd As Date) |
| GetReportComments | 27 - 146 | '+ ' Procedure : GetReportComments ' Comments : Producte a list (table) of all Printable comments that ' : fit the product /methods selected and the date range. ' Parameters: intReportType - 1 Consumption only, 2 Stock Status ' dtmStart ' dtmEnd - ' Created : 16-Feb-00 Jeff Leiner ' Modified : 26-Jan-04 LBlanken fix comment for stock '- |
| basResize | (General Declarations) | 1 - 11 |  |
| ResizeReportControls | 12 - 62 |  |
| basSafeWrappers | (General Declarations) | 1 - 13 |  |
| SafeClng | 111 - 123 |  |
| SafeCStr | 42 - 54 |  |
| SafeCVDate | 89 - 101 |  |
| SafeFormat | 14 - 30 |  |
| SafeNulltoString | 134 - 146 |  |
| SafeNZ | 158 - 170 |  |
| SafeTrim | 183 - 201 |  |
| SafeUCase | 66 - 78 |  |
| basSaveFile | (General Declarations) | 1 - 82 | '+ ' Module : basSaveFile ' Description: ' Procedures : ConvertTSB2Win(TSB\_Struct As TSBAPI\_OPENFILE, Win\_Struct As TSBAPI\_WINOPENFILENAME) ' ConvertWin2TSB(Win\_Struct As TSBAPI\_WINOPENFILENAME, TSB\_Struct As TSBAPI\_OPENFILE) ' CreateFilterString\_TSB(ParamArray varFilt() As Variant) ' GetSaveFile\_TSB(strInitialDir As String, strTitle As String, strDefName As String) ' RemoveNulls\_TSB(strIN As String) |
| ConvertTSB2Win | 90 - 131 | '+ ' Procedure : ConvertTSB2Win ' Comments : Converts the passed TSBAPI structure to a Windows structure ' Parameters: TSB\_Struct - record of type TSBAPI\_OPENFILE ' Win\_Struct - record of type TSBAPI\_WINOPENFILENAME ' Modified : 12 May 2004 LKB '- |
| ConvertWin2TSB | 141 - 156 |  |
| CreateFilterString\_TSB | 169 - 203 |  |
| GetSaveFile\_TSB | 215 - 264 |  |
| GetSaveFileXML\_TSB | 275 - 324 |  |
| RemoveNulls\_TSB | 332 - 354 | '+ ' Procedure : RemoveNulls\_TSB ' Comments : Removes terminator from a string ' Parameters: strIN - string to modify ' Returns : String - modified string ' Modified : 12 May 2004 LKB ' - |
| basShortCuts | (General Declarations) | 1 - 45 | '+ ' Module : basShortCuts ' Description: ' Procedures : CopyAppShortcut() ' GetSpecialFolder(CSIDL As Long) |
| CopyAppShortcut | 53 - 98 | '+ ' Procedure : CopyAppShortcut ' Comments : ' Parameters: - ' Returns : - ' Modified : 12 May 2004 LKB ' - |
| GetSpecialFolder | 108 - 141 |  |
| basStacks | (General Declarations) | 1 - 23 | '+ ' Module : basStacks ' Description: Contains functions for the creation of the stack used ' in tracking the progression of open forms in the database. ' Procedures : StackForms(strForm As String) ' UnstackForms() ' Created : 02 Apr 1999 Admin ' Modified : 23 Jul 1999 Admin Cleaned with CodeTools ' 30 May 2003 LBlanken as per code review ' 16 Jun 2003 LBlanken '- |
| JumpForms | 150 - 182 |  |
| ResetSubform | 33 - 89 | '+ ' Procedure : ResetSubForm ' Comments : Used to open form as sfrSform on frmMain ' Used when double clicking item in report data ' Parameters: strForm - Form to set sfrSform to ' Returns : - ' Modified : 30 May 2003 LBlanken as per code review ' 16 Jun 2003 LBlanken ' - |
| StackForms | 101 - 141 |  |
| UnstackForms | 193 - 219 |  |
| basStyle | (General Declarations) | 1 - 14 | '+ ' Module : basStyle ' Description: ' Procedures : SetStyle(lngStyle As Long, ctlTemp As Control) |
| SetStyle | 25 - 106 | '+ ' Procedure : SetStyle ' Comments : Sets the style based on table for forms and reports ' Parameters: lngStyle - unique identifier for style ' ctlTemp - control to apply style to ' Returns : - ' Modified : 30 May 2003 LBlanken as per code review ' 16 Jun 2003 LBlanken ' 18 AUG 2009 Jleiner - Added New Hightlighted Types ' - |
| basToolbarFunctions | (General Declarations) | 1 - 49 | '+ ' Module : basToolbarFunctions ' Description: ' Procedures : AddNewMB() ' CPTExport() ' DirtyContinue() ' FileClose() ' FileCopy\_PL() ' FileMenu(p\_strSelect As String) ' FileNew() ' FileOpen() ' ImportShipmentData() ' IsImportVisible() ' MakeCurrent() ' SampleMenuDisable() ' SetFormMode() ' SetProgramText() ' ToggleClickMe() ' UpdateBackEnd() ' MakeActive(p\_fActive As Boolean, p\_strCode As String) |
| AddNewMB | 59 - 142 | '+ ' Procedure : AddNewMB ' Comments : ' Parameters: - ' Modified : 14 Oct 2003 LKB '- '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ' This procedure creates a new menu bar. '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
| ChangeProgram | 153 - 196 |  |
| CPTExport | 205 - 222 |  |
| DirtyContinue | 232 - 260 |  |
| FileClose | 270 - 343 |  |
| FileCopy\_PL | 353 - 439 |  |
| FileMenu | 449 - 495 |  |
| FileNew | 505 - 531 | 'check if form is dirty and prompt to save (false only if user says cancel) |
| FileOpen | 542 - 866 |  |
| HelpClose\_TSB | 895 - 915 | ' Comments : marks the specified help file as closed ' Parameters: strFile - name of help (\*.HLP) file to mark ' Returns : True if successful, False otherwise ' |
| HelpContext\_TSB | 877 - 894 |  |
| ImportShipmentData | 924 - 945 |  |
| IsImportVisible | 957 - 988 |  |
| LangMenus | 999 - 1306 |  |
| lblID | 1316 - 1330 |  |
| MakeActive | 2039 - 2080 |  |
| MakeCurrent | 1340 - 1425 |  |
| OpenAbout | 1435 - 1447 |  |
| OpenHelp | 1457 - 1469 |  |
| OpenLanguage | 1479 - 1496 |  |
| OpenProperties | 1506 - 1518 |  |
| RelinkToDefault | 1528 - 1562 |  |
| RunLanguageForm | 1573 - 1627 |  |
| SampleMenuDisable | 1642 - 1655 |  |
| SaveSubForm | 1665 - 1705 |  |
| SetFormMode | 1716 - 1743 |  |
| SetProgramText | 1753 - 1793 |  |
| ToggleClickMe | 1808 - 1828 |  |
| UpdateAndMove | 1838 - 1878 |  |
| UpdateBackEnd | 1889 - 1927 |  |
| UpdateCommandBars | 1937 - 2003 |  |
| UpdateTreeviewLanguage | 2013 - 2028 |  |
| basTranslation | (General Declarations) | 1 - 20 | '+ ' Module : basTranslation ' Description: ' Procedures : BuildTranslationTable() ' CaptureTabPageCaption() ' CreateCaptiontext() ' CreateFormTitleCaptions() ' CreateReportCaptiontext() ' CreateStatusbartext() ' CreateToolTipText() ' setButtonColor() |
| 1 | 92 - 111 |  |
| 1 | 149 - 167 |  |
| 1 | 216 - 232 |  |
| 1 | 273 - 291 |  |
| 2 | 387 - 406 |  |
| 3 | 443 - 456 |  |
| 4 | 329 - 348 |  |
| BuildTranslationTable | 28 - 55 | '+ ' Procedure : BuildTranslationTable ' Comments : build the translation table based on forms ' Parameters: - ' Returns : - ' Modified : 12 May 2004 LKB ' - |
| setButtonColor | 465 - 502 |  |
| basTreeviewUtils | (General Declarations) | 1 - 15 | '+ ' Module : basTreeviewUtils ' Description: ' Procedures : CreateTreeview() ' SetActiveNode(strKey As String) |
| CreateParenttoRoot | 26 - 106 | '+ ' Procedure : CreateParenttoRoot ' Comments : For a given nodeID in the treeview, make sure it parent ' : nodes exist all to the root (ParentID = 0). This procedure ' : makes Recursive calls to itself. ' Parameters: lngNodeID - The parent node to test for ' : ctlTreeview - The Treeview being populated ' Returns : Boolean - True returns that the node exists. ' Modified : ' - |
| CreateTreeview | 116 - 174 |  |
| SetActiveNode | 183 - 201 |  |
| basUndoMainSubform | (General Declarations) | 1 - 20 | '+ ' Module : basUndoMainSubform ' Description: ' Procedures : SaveMain(Frmin As Form, strTempTable As String) ' SaveSub(Frmin As Form, strTempTable As String) ' UndoMain(Frmin As Form, strTempTable As String) ' UndoSub(Frmin As Form, strTempTable As String) |
| SaveMain | 29 - 83 | '+ ' Procedure : SaveMain ' Comments : Save values in main form to temporary table ' Parameters: Frmin - the name of the form ' strTempTable - the name of the temporary table ' Modified : 30 May 2003 LBlanken as per code review ' 16 Jun 2003 LBlanken '- |
| SaveSub | 94 - 155 |  |
| UndoMain | 166 - 217 |  |
| UndoSub | 229 - 325 |  |
| basUpgradeBackEnd | (General Declarations) | 1 - 18 | '+ ' Module : basUpgradeBackEnd ' Description: ' Procedures : checkBackEndVersion(strDataDir As String) ' CopyIndexes\_TSB(strDatabase As String, strDestination As String, strTable As String) ' CopyRelations\_TSB(strDatabase As String, strDestination As String) ' CreateAttachement\_TSB(strSourceDatabase As String, strDestDatabase As String, strTable As String) ' DeleteRelations\_TSB(strDestination As String) ' UpgradeProgram(sProgram As String, sDir As String) |
| checkBackEndVersion | 31 - 107 | '+ ' Procedure : checkBackEndVersion ' Comments : Compares the version number of the PMP\_MPTY against ' : that of the globalmoh.MDB being checked. If the ' : files is found and the version numbers match then ' : the version is OK, else the files will need to be ' : updated. ' Parameters: strDataDir - ' Returns : Integer - ' Created : 06/04/98 JSL ' Modified : 12 May 2004 LKB ' - |
| CopyIndexes\_TSB | 120 - 296 |  |
| CopyRelations\_TSB | 307 - 400 |  |
| CreateAttachement\_TSB | 412 - 448 |  |
| DeleteRelations\_TSB | 458 - 517 |  |
| UpgradeProgram | 529 - 755 |  |
| basUpgradeMDB\_Utils | (General Declarations) | 1 - 25 | '+ ' Module : basUpgradeMDB\_Utils ' Description: ' Procedures : ClearOldObjects(strUserFile As String) ' CreateUpgradeMDB(sPath As String) ' UpdateProgramTable() |
| ClearOldObjects | 26 - 123 |  |
| CreateUpgradeMDB | 134 - 192 |  |
| UpdateProgramTable | 204 - 289 |  |
| basWindowsVersion | (General Declarations) | 1 - 20 |  |
| fOSNameXP | 21 - 36 |  |
| basXMLConsumptionImport | (General Declarations) | 1 - 15 | '+ ' Module : basXMLConsumptionImport ' Description: ' Procedures : ConsDataExists() ' ConsImport() ' OpenImport() ' OpenReconciliation() |
| ConsDataExists | 23 - 39 | '+ ' Procedure : ConsDataExists ' Comments : checks if data exists in import tables ' Parameters: - ' Returns : Boolean - ' Modified : 21 Aug 2003 LKB ' - |
| ConsImport | 49 - 82 |  |
| OpenImport | 92 - 115 |  |
| OpenReconciliation | 125 - 142 |  |
| basXMLDump | (General Declarations) | 1 - 8 |  |
| CreateXMLFromTables | 15 - 52 | '+ ' Procedure : CreateXMLFromTables ' Comments : Create the Raw XML Dump based on the file ' Parameters : - ' Modified : 22 Mar 2007 JSL '- |
| ExportProgramDataXML | 61 - 115 |  |
| MemotoString | 125 - 138 |  |
| TransformXML | 147 - 175 |  |
| basXMLForecastImport | (General Declarations) | 1 - 15 | '+ ' Module : basXMLForecastImport ' Description: ' Procedures : ConsDataExists3() ' ConsImport3() ' OpenImport3() ' OpenReconciliation3() |
| ClearForecastData | 175 - 196 |  |
| ConsDataExists3 | 23 - 41 | '+ ' Procedure : ConsDataExists3 ' Comments : checks if data exists in import tables ' Parameters: - ' Returns : Boolean - ' Modified : 29 Sep 2006 MAT ' - |
| ConsImport3 | 51 - 91 |  |
| OpenImport3 | 101 - 132 |  |
| OpenReconciliation3 | 142 - 167 |  |
| basXMLProductImport | (General Declarations) | 1 - 15 | '+ ' Module : basXMLProductImport ' Description: ' Procedures : ConsDataExists2() ' ConsImport2() ' OpenImport2() ' OpenReconciliation2() |
| ClearSCMSData | 151 - 178 |  |
| ConsDataExists2 | 23 - 43 | '+ ' Procedure : ConsDataExists2 ' Comments : checks if data exists in import tables ' Parameters: - ' Returns : Boolean - ' Modified : 24 Jul 2006 MAT ' - |
| ConsImport2 | 53 - 83 |  |
| InSubList | 186 - 210 | ' ---------------------------------------------- ' Creates a delimited string of values, converts ' it to an array and determines if the array ' contents contain the specified string ' ============================================== |
| OpenImport2 | 93 - 116 |  |
| OpenReconciliation2 | 126 - 143 |  |
| basXMLShipmentImport | (General Declarations) | 1 - 16 | '+ ' Module : basXMLShipmentImport ' Description: ' Procedures : ShipDataExists() ' ShipImport() ' OpenImportShipment() ' OpenReconciliationShipment() |
| ClearShipmentData | 153 - 181 |  |
| OpenImportShipment | 95 - 118 |  |
| OpenReconciliationShipment | 128 - 145 |  |
| ShipDataExists | 24 - 45 | '+ ' Procedure : ShipDataExists ' Comments : checks if data exists in import tables ' Parameters: - ' Returns : Boolean - ' Modified : 1 Sep 2009 mahmed ' - |
| ShipImport | 55 - 85 |  |
| clsCloneProduct | (General Declarations) | 1 - 7 |  |
| CloneConsumption | 197 - 233 |  |
| createCloneTable | 126 - 184 |  |
| ProduceConsumptionRecords | 246 - 322 |  |
| ProductID | 36 - 47 | '+ ' Property(S): ProductID ' Comments : ' Parameters : ' Created : 16 Jul 2009 jleiner ' Modified : '- |
| ProductID | 16 - 28 | '+ ' Property(G): ProductID ' Comments : ' Parameters : ' Returns : String ' Created : 16 Jul 2009 jleiner ' Modified : '- |
| setStartDate | 99 - 110 |  |
| StartDate | 77 - 88 |  |
| StartDate | 57 - 67 |  |
| clsCloseCommand | (General Declarations) | 1 - 46 | '+ ' Module : clsCloseCommand ' Description: Control the Access close button. ' When previewing reports, it is very easy for the user to close PipeLine while ' only attempting to close the previewed report and return to PipeLine. ' This occurs because the user confuses the Red X button in the upper ' right corner of the screen for the command "close report" ' when it actually executes the command "exit PipeLine". |
| Enabled | 64 - 78 |  |
| Enabled | 47 - 62 |  |
| clsConsumptionDataXML | (General Declarations) | 1 - 45 | '+ ' Module : clsConsumptionDataXML ' Description: Object for importing consumption data in Supply Chain Manager ' : XML format ' Procedures : Import() ' LoadXML(strFile As String) ' Get DataInterval() ' Let DataInterval(ByVal dblValue As Double) ' Get DataSource() ' Let DataSource(ByVal strName As String) ' Get DateExported() ' Let DateExported(ByVal dtmValue As Date) ' Get EndPeriod() ' Let EndPeriod(ByVal dtmValue As Date) ' Get StartPeriod() ' Let StartPeriod(ByVal dtmValue As Date) ' Get SystemName() ' Let SystemName(ByVal strName As String) |
| DataInterval | 260 - 272 |  |
| DataInterval | 239 - 251 |  |
| Datasource | 303 - 315 |  |
| Datasource | 282 - 294 |  |
| DateExported | 346 - 358 |  |
| DateExported | 325 - 337 |  |
| EndPeriod | 389 - 401 |  |
| EndPeriod | 368 - 380 |  |
| Import | 46 - 155 |  |
| LoadXML | 167 - 229 |  |
| StartPeriod | 432 - 444 |  |
| StartPeriod | 411 - 423 |  |
| SystemName | 454 - 466 |  |
| SystemName | 475 - 487 |  |
| clsExportExcel | (General Declarations) | 1 - 46 |  |
| Class\_Initialize | 553 - 556 |  |
| Class\_Terminate | 557 - 568 |  |
| CloseFile | 128 - 133 |  |
| ColumnLetter | 260 - 279 |  |
| CreateFile | 65 - 90 |  |
| CreateWorksheet | 143 - 149 |  |
| deleterows | 117 - 120 |  |
| FormatCells | 300 - 359 |  |
| FormatSelection | 445 - 485 |  |
| FormatWorksheet | 214 - 229 |  |
| getcolumn | 282 - 285 |  |
| getexcelfilename | 47 - 64 |  |
| GetLastActiveColumn | 368 - 375 |  |
| GetLastActiveRow | 360 - 367 |  |
| GetValue | 288 - 290 |  |
| InsertRows | 376 - 381 |  |
| MergeCells | 291 - 299 |  |
| MoveActiveSheetToEnd | 538 - 540 |  |
| OpenFile | 121 - 127 |  |
| PageSetup | 396 - 444 |  |
| PutValueN | 105 - 116 |  |
| PutValueS | 100 - 104 |  |
| RemoveEmptySheets | 542 - 552 |  |
| RemoveEqualValues | 486 - 537 |  |
| RemoveWorksheets | 150 - 156 |  |
| RenameWorkSheet | 157 - 209 |  |
| SaveFile | 135 - 140 |  |
| SelectRange | 382 - 395 |  |
| SelectWorksheet | 210 - 213 |  |
| setcolumn | 257 - 259 |  |
| setcolwidth | 91 - 95 |  |
| setrowcol | 96 - 99 |  |
| SetValue | 230 - 256 |  |
| clsForecastDataXML | (General Declarations) | 1 - 50 | '+ ' Module : clsForecastDataXML ' Description: Object for importing consumption data from Quantimed/SCMS ' : XML format ' Procedures : Import() ' LoadXML2(strFile As String) ' Get DataInterval() ' Let DataInterval(ByVal dblValue As Double) ' Get DataSource() ' Let DataSource(ByVal strName As String) ' Get DateExported() ' Let DateExported(ByVal dtmValue As Date) ' Get EndPeriod() ' Let EndPeriod(ByVal dtmValue As Date) ' Get StartPeriod() ' Let StartPeriod(ByVal dtmValue As Date) ' Get SystemName() ' Let SystemName(ByVal strName As String) |
| DataInterval | 407 - 419 |  |
| DataInterval | 428 - 440 |  |
| Datasource | 450 - 462 |  |
| Datasource | 471 - 483 |  |
| DateExported | 493 - 505 |  |
| DateExported | 514 - 526 |  |
| EndPeriod | 557 - 569 |  |
| EndPeriod | 536 - 548 |  |
| Import | 51 - 227 |  |
| LoadXML2 | 239 - 397 |  |
| StartPeriod | 579 - 591 |  |
| StartPeriod | 600 - 612 |  |
| SystemName | 622 - 634 |  |
| SystemName | 643 - 655 |  |
| clsProductDataXML | (General Declarations) | 1 - 34 | '+ ' Module : clsProductDataXML ' Description: Object for importing product data from SCMS source into PL ' : XML format ' Procedures : Import() ' LoadXML(strFile As String) |
| Import | 35 - 410 |  |
| ImportCountries | 600 - 639 |  |
| LoadXML | 422 - 598 |  |
| clsShipmentDataXML | (General Declarations) | 1 - 28 | '+ ' Module : clsShipmentDataXML ' Description: Object for importing product data from SCMS source into PL ' : XML format ' Procedures : Import() ' LoadXML(strFile As String) |
| checkDuplicateMapping | 533 - 581 |  |
| Import | 29 - 98 |  |
| ImportDatasources | 233 - 272 |  |
| ImportFundingsources | 274 - 313 |  |
| ImportProducts | 315 - 379 |  |
| ImportShipments | 382 - 527 |  |
| ImportSuppliers | 182 - 231 |  |
| LoadXML | 108 - 180 |  |
| modEstimation | (General Declarations) | 1 - 15 | '+ ' Module : modEstimation ' Description: ' Procedures : creaMonthlyTrendConsumption(strProductID As String, intEndYear As Integer) ' GetPeriodAverages(dtmStart As String, dtmEnd As String, blnUseEstimates As Integer) ' TempAttach(strDatabase) |
| creaMonthlyTrendConsumption | 32 - 300 | '+ ' Procedure : creaMonthlyTrendConsumption ' Comments : Create MonthlyConsumption table for strProductID, ' : ending in intEndYear. First creates two temporary files from ' : data in table Consumption. One file contains monthly estimates ' : for actual consumption. One file contains monthly estimates for ' : forecast consumption. The monthly consumption is the amount ' : consumed averaged over the number of months; the last month in ' : the period absorbs any rounding errors. Finally, merge the two ' : files, with actual consumption, where available, taking precedence ' : over forecast. ' Parameters: strProductID ' intEndYear - ' Returns : - ' Modified : 01 Jul 1998 JSL ' - |
| GetPeriodAverages | 314 - 544 |  |
| GetPeriodAveragesForecast | 557 - 789 |  |
| TempAttach | 799 - 823 |  |
| modHelpContextIDs | (General Declarations) | 1 - 13 | ' Module : modHelpContextIDs ' Description: ' Procedures : CreateHelpContextIDs() ' CreateHelpContextIDsReports() ' GetHelpID(strForm As String, fIsForm As Boolean, bytIndex As Byte) |
| CreateHelpContextIDs | 14 - 55 | ' Comments : Generate a table of the HelpContext IDs text for all form ' : Objects ' Parameters : None ' Returns : None ' Created : 03/23/2000 JSL '----------------------------------------------------------------------- |
| CreateHelpContextIDsReports | 57 - 99 | ' Comments : Generate a table of the HelpContext IDs text for all Report ' : Objects ' Parameters : None ' Returns : None ' Created : 03/23/2000 JSL '----------------------------------------------------------------------- |
| GetHelpID | 101 - 156 | ' Comments : ' Parameters : ' Returns : String ' Created : 02-Aug-00 Jeff Leiner ' Modified : ' ' -------------------------------------------------------- |
| modInterpolation | (General Declarations) | 1 - 14 | '+ ' Module : modEstimation ' Description: ' Procedures : creaMonthlyTrendConsumption(strProductID As String, intEndYear As Integer) ' GetPeriodAverages(dtmStart As String, dtmEnd As String, blnUseEstimates As Integer) ' TempAttach(strDatabase) |
| Interpolate | 15 - 59 |  |
| Table Initialization | (General Declarations) | 1 - 26 | '+ ' Module : Table Initialization ' Description: ' Procedures : AttachData(intStartup As Integer) ' AttachProgram() ' CheckProductDSLMonths() ' CheckY2kDates() ' CopyFile(strSource As String, strDestination As String) ' CopyFile\_TSB(strSource As String, strDestination As String) ' GetAttachedPath(strDatabase As String, strTable As String) ' GetDBUpdateDate() ' iniTableLang() ' KillFile\_tsb(strFile As String) ' RemoveOldVersion() ' SetLanguage(strLanguage As String) ' StartUp() ' UpdateShipData() |
| AttachData | 43 - 308 | '+ ' Procedure : AttachData ' Comments : ' Update connection information to reflect FP program that is active, ' as indicated by IsCurrent setting in table Program. ' If intStartUp is true, error returns invoke form frmProgram, ' which allows the user to reconfigure. ' If new tables are added to the program, or attached, side of the tables, ' they must be added to this routine. ' Tables used to hold results for display of printing are cleared, ' to eliminate any references to installation specific lookup variables ' Parameters: intStartup - ' Returns : Integer - ' Modified : 01 Jul 1998 JSL '12 May 2004 LKB ' - |
| AttachProgram | 319 - 366 |  |
| CheckProductDSLMonths | 377 - 426 |  |
| CheckY2kDates | 439 - 503 |  |
| CopyFile | 514 - 529 |  |
| CopyFile\_TSB | 540 - 582 |  |
| GetAttachedPath | 593 - 625 |  |
| GetDBUpdateDate | 636 - 652 |  |
| iniTableLang | 663 - 843 |  |
| KillFile\_tsb | 853 - 868 |  |
| RemoveOldVersion | 878 - 900 |  |
| SetLanguage | 911 - 980 |  |
| StartUp | 990 - 1022 |  |
| UpdateShipData | 1032 - 1047 |  |
| Table Utils | (General Declarations) | 1 - 34 | '+ ' Module : Table Utils ' Description: ' Procedures : AtNewRecord(frm As Form) ' CodeUsed(strCode As String, strTable As String, strCodeField As String) ' CreateMonthlyAction(intEndYear As Integer) ' CreateMonthlyConsumption(strProductID As String, intEndYear As Integer, intKeepData As Integer) ' CreateMonthlyMethodConsumption(varStrMethod, varIntEndYear) ' CreateMonthlyMethodStock(strMethod As String, intEndYear As Integer, Optional fSkipPlanned As Boolean) ' CreateMonthlyProblem(intEndYear As Integer) ' CreateMonthlyShipment(strProductID As String, Optional fSkipPlanned As Boolean) ' CreateMonthlyStock(strProductID As String, ByVal intEndYear As Integer, intKeepData As Integer, Optional fSkipPlanned As Boolean) ' CreateShipSched(intEndYear As Integer) ' GetDesLevel(varProductID) ' GetDesMin(varProductID) ' GetLastYear(strProductID As String) ' getLeadTime(strProductID As String, strSupplierID As String, intLTType As Integer) ' getmaxMonths(strProductID As String) ' GetMethodActualorForecast(intYear As Integer, strPeriod As String, intMonth As Integer) ' getminMonths(strProductID As String) ' getParmValue(varParm) ' SetCentury(vardate) ' SetParm(strPArm As String, strValue As String) |
| AtNewRecord | 44 - 57 | '+ ' Procedure : AtNewRecord ' Comments : Returns true if record set in frmForm is positioned at new record ' : Reads bookmark; if recordset is at new record, Access generates an error return ' Parameters: frm - ' Returns : - ' Modified : 01 Jul 1998 JSL ' 20 Jun 2003 LBlanken ' - |
| CodeUsed | 71 - 126 |  |
| CreateMonthlyAction | 144 - 371 |  |
| CreateMonthlyConsumption | 391 - 716 |  |
| CreateMonthlyMethodConsumption | 726 - 775 |  |
| CreateMonthlyMethodStock | 788 - 880 |  |
| CreateMonthlyProblem | 896 - 1103 |  |
| CreateMonthlyShipment | 1120 - 1219 |  |
| CreateMonthlyStock | 1240 - 1663 |  |
| CreateShipSched | 1681 - 1874 |  |
| GetDesLevel | 1887 - 1918 |  |
| GetDesMin | 1931 - 1983 |  |
| GetLastYear | 1996 - 2048 |  |
| getLeadTime | 2067 - 2136 |  |
| getmaxMonths | 2148 - 2169 |  |
| GetMethodActualorForecast | 2183 - 2247 |  |
| getminMonths | 2259 - 2280 |  |
| getParmValue | 2291 - 2311 |  |
| GetProductID | 2321 - 2333 |  |
| SetCentury | 2345 - 2399 |  |
| SetParm | 2412 - 2436 |  |

Menu Structure

Overview

PipeLine contains a dynamic menu structure. The menu is created each time the application is open and when the language is changed. The code that is ran to create the menu bar can be found in AddNewMB which calls the LangMenus in the basToolbarFunctions module. This code sets the OnAction property for the menu item along with the help file and context ID. In some cases, the OnAction code calls other modules and in other cases it calls macros. The following table lists the menu item and the OnAction command it calls.

Program calls by Menu choices

Table - Program Calls by Menu Choices

| Menu Option | Sub Menu | | OnAction command | Description |
| --- | --- | --- | --- | --- |
| File | New | | =FileMenu(""new"") | Click to create New PipeLine Data files |
| Open | | =FileMenu(""Open"") | Click to open existing PipeLine Data file |
| Copy | | =FileMenu(""Copy"") | Click to copy current data file to a new location |
| Close | | =FileMenu(""Close"") | Deactivate current data file |
| Properties | | =OpenProperties() | Open properties form |
| Exit | | =FileMenu(""Exit"") | Exit PipeLine |
| Import | Products | New | mcrProduct.ImportNew | Import new product file |
| Update | mcrProduct.ImportUpdate | Updates existing products that match files in product file |
| Consumption | Forecast | mcrConsumption.ImportForecast | Import forecast data from Quantimed |
| Actual | mcrConsumption.Import | Imports actual data from Supply Chain Manager |
| Reconcile | mcrConsumption.Reconcile | Reconciles last Supply Chain Manager import |
| Shipments | Initial | mcrShipments.ImportPipleLine |  |
| PipeLine | mcrMenu.ToolsImportShipments | Import shipments |
| Export | Program Data | | mcrMenu.ExportData | Exports all data |
| Shipments | | mcrMenu.ExportData | Exports all shipment data |
| Tools | Language | English | mcrmenu.LanguageE | Changes display language to English |
| Português | mcrmenu.LanguageP | Changes display language to Portuguese |
| Français | mcrmenu.LanguageF | Changes display language to French |
| Español | mcrmenu.LanguageS | Changes display language to Spanish |
| Tools *(Cont.)* | PipeLine Summary | | mcrMenu.PLSummary | Opens PipeLine Summary |
| Compact Backend | | mcrMenu.CompactBE | Compacts current data file |
| Choose PDF Printer | | mcrmenu.ChoosePDF | Opens form to select PDF printer |
| Window | *<various>* | | =ChangeProgram('" & rstPrograms!ProgramName & "') | Opens activated program files |
| Help | Help | | =OpenHelp() | Opens help file |
| About PipeLine | | =OpenAbout() | Opens PipeLine about form |

Archiving and Backup

Overview

Currently there is not an archiving function built in to PipeLine. One of the benefits however is that the backend database can be copied and saved at any time (better to be done while PipeLine is closed). Users can also utilize the copy function found in the File menu to easily copy the backend file to another filename for backup purposes.

It is also recommended that the backend database file compacted and repaired at regular intervals.

Interfaces with Other Systems

Overview

PipeLine allows the user to interface with generic systems through the export of all program data, export of shipment data, and import of product, consumption, and shipment data via an xml file format.

Generic systems (Export of all program data)

PipeLine can interface with generic systems through the exporting of an xml file. The xml is expected to follow a basic schema. The schema is explained in the PipelineXMLOutputSchema\_070924.xsd file found on the developer source code cd in the /schemas folder.

Figure - xml Export file schema diagram

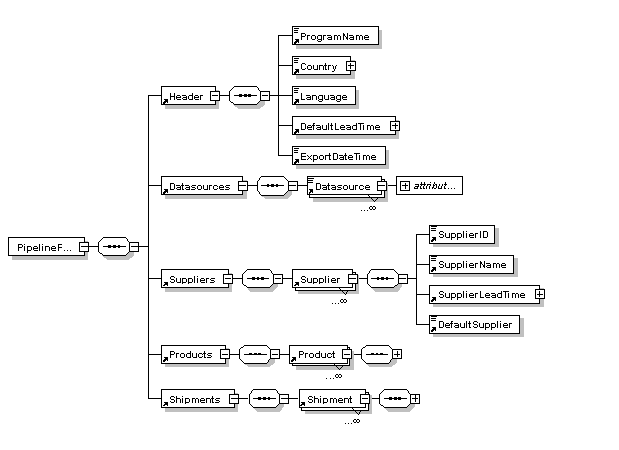


Figure - xml Export file schema diagram

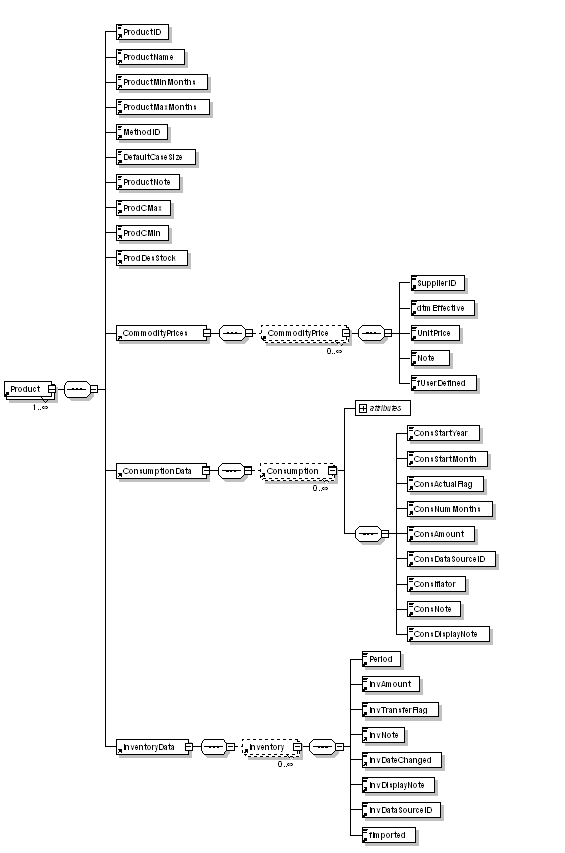
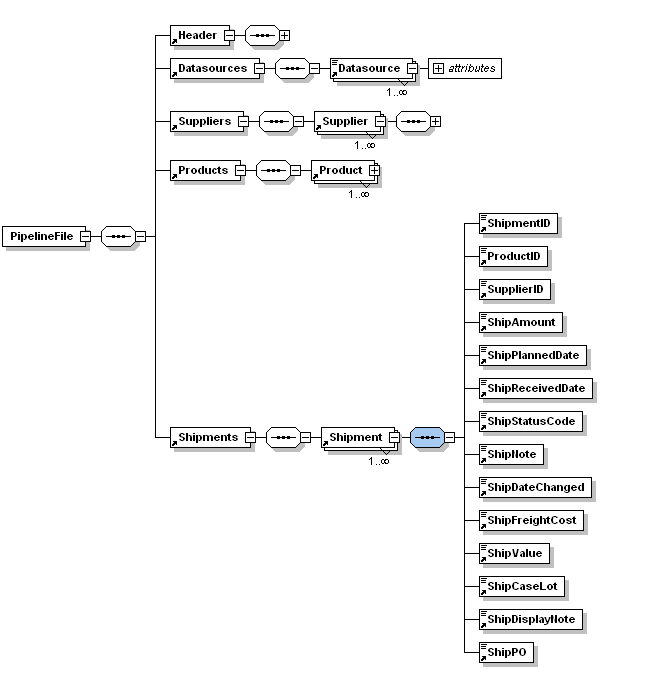


Figure - xml Export file schema diagram



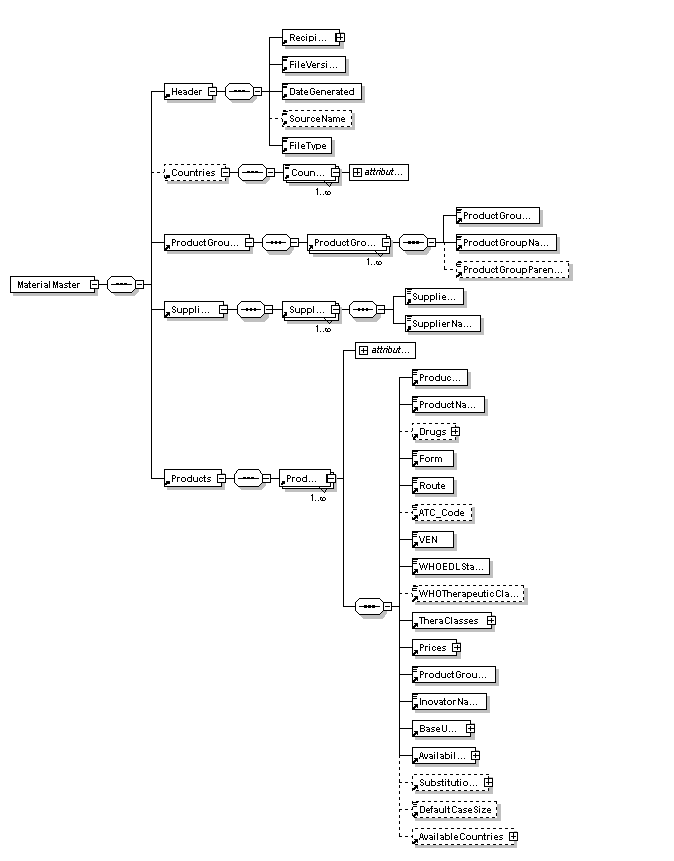
Other Pipeline Databases (import shipment data)

PipeLine can interface with other PipeLine database through the importing of an xml file. The xml is expected to follow the schema outlined for generic system above.

E-Catelog (Import product data)

PipeLine can interface with E-Catelog (or other product list sources) through the importing of an xml file. The xml is expected to follow a basic schema. The schema is explained in the MaterialMasterNorm\_070516\_NoCustomType.xsd file found on the developer source code cd in the /schemas folder.

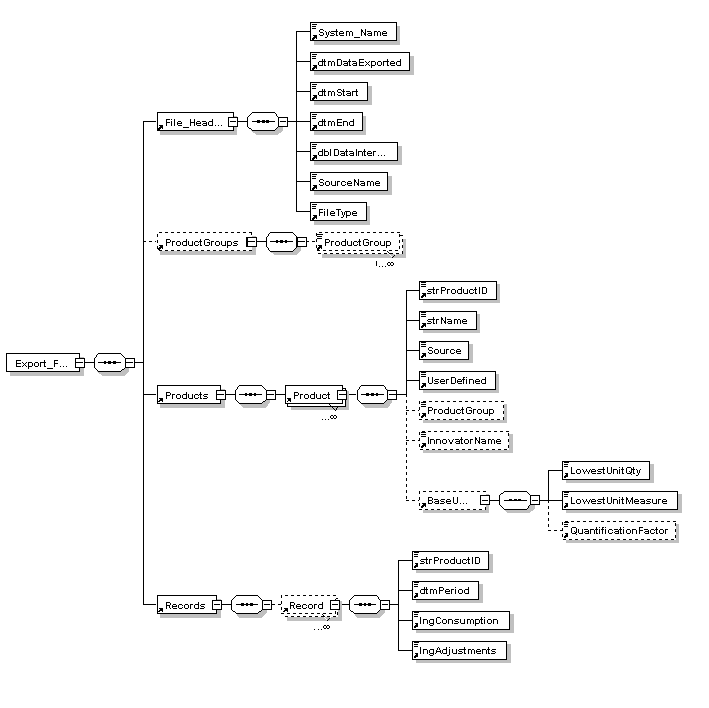
Figure - xml Export file schema diagram



Quantimed (import forecast consumption data)

PipeLine can interface with Quantimed through the importing of an xml file. The xml is expected to follow a basic schema. The schema is explained in the QuantimedForecastOutput.xsd file found on the developer source code cd in the /schemas folder.

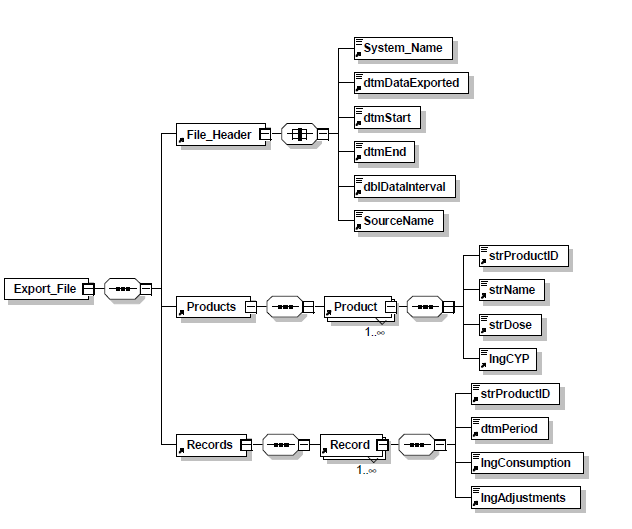
Figure - xml Export file schema diagram



Supply Chain Manager (import actual consumption data)

PipeLine can interface with Supply Chain Manager through the importing of an xml file. The xml is expected to follow a basic schema. The schema is explained in the SCMgr\_PipeLine\_Export.xsd file found on the developer source code cd in the /schemas folder.

Figure : xml Export file schema diagram



APPENDIX A: UPGRADING DATA FILES

Overview

When upgrading PipeLine sometimes changes need to occur in the backend data files. In PipeLine a mechanism has been created that will easily allow the system to do this automatically when opening a previous versions file. Basically, the application will determine which version the data file is, will create a new file as specified by the user, and then run the appropriate SQL statements to copy the data into the file.

Version Numbering

PMP\_MPTY.MDB is the empty datafile for PipeLine. When creating a new backend, PipeLine copies this file to the specified location and adds the program data to it. The table, tblBE\_Version, in this file contains the version number of the backend along with the date that the file was updated. When PipeLine opens a new datafile, it verifies this number with the number stored in the backend that it is opening. (See code: CheckBackEndVersion). If the version numbers are different then the upgrade procedure is called. If the numbers are the same then the file will open and the user is taken to the main menu. Please note that when making structure changes to the data is is imperative that the changes be made in the the PMP\_MPTY.MDB file and that this version number is updated.

Upgrade Procedure

When the version number in the backend that the user is opening does not match the number in the PMP\_MPTY.MDB file, the upgrade procedure is called (see code: UpgradeProgram). This procedure prompts the user for a new file name and location for the upgraded datafile. It then creates a log file to keep track of all changes and errors that occur during the upgrade. The procedure will copy the PMP\_MPTY.MDB to location specified and then opens the table tlkUpgrade and runs the appropriate sql statement. Each statement is created to copy the data from the existing backend into the new backend. This ensures that all the data gets placed in the correct location in the new backend. Once all the SQL statements are run, the procedure will change the name to the one specified and open the file. The user is then taken to the Main Menu. Please note again, that the original datafile remains intact during this procedure. This guarantees the user will not lose any data.

SQL Statements Used

The most complicated part of the upgrade procedure is writing the SQL statements. There are three types of SQL statements to be written. They are: Update, Append and Delete. Each table should have at least one of each of these but in some cases can have multiple Append and Update statements. In order to determine which statements to run, a version number is stored with each statement along with an order number. The order number specified the order in which the SQL statements are ran, and the version number dictates which code is ran depending on the version of the existing data backend. When updating the version number, the new upgrade version number is double the previous number. (For example, version 3.08 has a version number of 64. Version 3.09’s number would then be 128.) This ensures that numbers will not duplicate. To specify, which versions the SQL statement is ran for, the version number in the table is the sum of the version number is runs for. (And so, number 129 would run when the backend is version 3.09 AND 3.01.) With each release, the version numbers in the table must be updated along with each statement. The upgrade procedure will then determine which SQL statements to run based on this numbering system and the verstion number of the existing data. The delete statement is ran first so that non system data is deleted. The update statements are next. These will match any remaining system data and update as needed. And finally, the append statements will then copy in all other existing data. This procedure ensures that all data is updated to the new structure correctly.

APPENDIX B: PIPELINE SUMMARY

Overview

This option (only available in English) is intended for program managers and/or consultants responsible for managing multiple programs tracked by PipeLine. It aggregates PipeLine data from selected programs, and arranges the data so it can be presented in a selected graph or report.

The following graphs and reports are available:

**Shipment Costs by Supplier Report**This report is similar to the PipeLine Shipment Summary report. It displays shipment costs, including product and freight costs, for selected supplier/status of selected, aggregated programs.

**Shipment Orders Report**This report is similar to the PipeLine Shipment Orders report. It displays quantities and cost of orders for a selected supplier/status of selected, aggregated programs.

**Consumption Graph/Export**This graph is similar to the PipeLine Consumption graph. It produces a monthly or quarterly bar chart showing actual and forecast consumption by product or method of selected, aggregated programs. This option lets you export aggregated data.

**Couple-Years of Protection (CYP) Graph**This graph is similar to the PipeLine couple-years of protection (CYP) graph. It produces a monthly or quarterly bar chart showing actual and forecast CYP by product or method of selected, aggregated programs.

Access the PipeLine Summary Module

The PipeLine Summary Module is accessed through PipeLine.

From the Menu Bar—

1. Click on the Tools option of the Menu Bar to display the Tools pull-down menu.
2. Click on the PipeLine Summary pull-down menu option to display the PipeLine Summary menu screen.

Selecting Programs

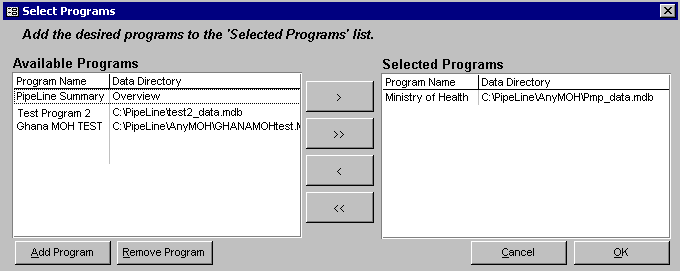
The Select Programs option lets you select the program(s) to aggregate. You must select a program or programs for aggregation before you create reports or graphs.

From the PipeLine Summary Menu—

1. Click on the Select Programs button.

PipeLine displays the Select Program screen.

Figure - Select Program



If the program (or programs) you need to aggregate does not appear in the list of available programs, you can add them.

Adding Programs

Use the Add Program button to add programs for selection and aggregation.

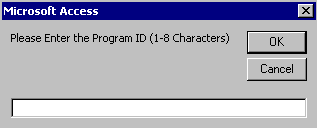
1. Click on the Add Programs button

PipeLine displays the Load Database window.

1. Locate the data file corresponding to the program you want to add.
2. Click on the program you need, then click on the Open button.

PipeLine displays a window so you can associate an eight-character ID with the selected program.

Figure - Enter Program ID window



1. Type the program's eight-character ID.
2. Click on the OK button to save the ID, and close the Program ID window.

The selected program is displayed in the Available Programs window, on the Select Program screen.

1. Repeat the process until you have added all the programs you need.

After your list is complete, you must select the programs you need to aggregate for reporting. This can be done in two ways:

1. pipeline pics\get-2-but.bmpClick on the Select All button to move the programs in the Available Programs window to the Selected Programs window  
   Or, click on the program you need, and click on the Select button to move it from the Available Programs window to the Selected Program window.



1. Repeat, as needed.

After you select the programs you need—

Changing Data

If you make changes to a program's data in PipeLine, you must reload the program into PipeLine Summary to ensure that the changes are reflected in the summary reports and graphs.

1. Click on the OK button to aggregate the data and return to the PipeLine Summary menu.

Creating Reports and Graphs

PipeLine can create summary reports and graphs.

Shipment Costs by Supplier Report

The Shipment Costs by Supplier report groups shipment quantities and costs by supplier, method, product, program, and status.

This is useful if you have shipments going to multiple programs from the same supplier, and you need to look at total shipments and associated costs for each supplier.

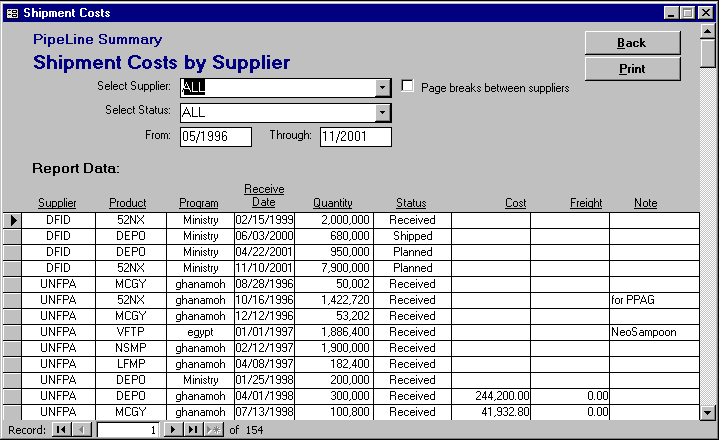
Use this report to review shipment information (including schedules and budgets) for a selected set of programs.

From the PipeLine Summary Menu—

1. Click on the Shipments by Supplier Report button.

PipeLine displays shipments and their associated costs. The Shipment Costs by Supplier screen is displayed after the report is created. Use this screen to create shipment cost reports for other suppliers.

Figure - — PipeLine Summary Shipment Costs by Supplier screen



By default, the Shipment Costs by Supplier report is set for all products and all shipment statuses, with a reporting period of 5 years. To limit the report's scope—

1. Click on the arrow next to the Select Supplier field, and select a supplier from the pull-down menu.
2. Click on the Select Status field and select the shipment status you want.
3. Click on the From field, and type the beginning date of the report.
4. Click on the Through field, and type the ending date of the report.

The Page Breaks Between Suppliers fields let you include page breaks between suppliers listed in the report (when the report is generated for more than one supplier).

1. Click on the Page Breaks Between Suppliers field to include page breaks between suppliers.

Shipment Order Report

The Shipment Order report groups shipments by status, supplier, method, and product. Shipments with status *planned*" are grouped under "New shipments requested this order." Shipments with status *shipped* or status *ordered* are grouped under "Confirmation of shipments previously ordered and expected." Shipments with status *received* are grouped under "Shipments received (in report range)."

This is useful if you have shipments to multiple programs and you need to look at which shipments need to be ordered or confirmed with the suppliers or donors.

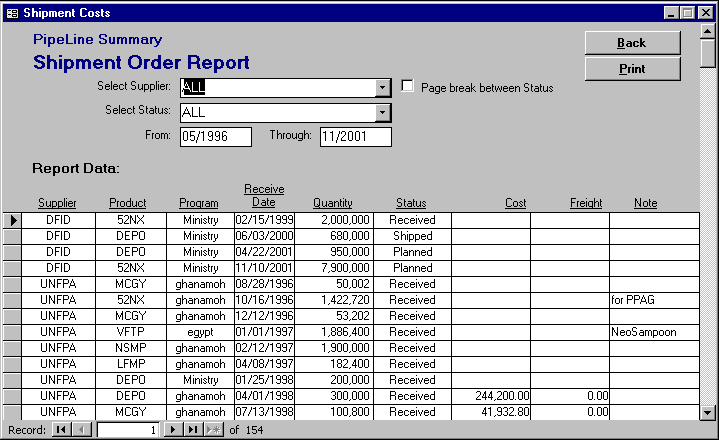
Use this report to review shipment status information (including schedules and costs) for a selected set of programs. Select a single supplier. Present this report to that supplier to order planned shipments, confirm previously ordered shipments, and confirm receipt of shipments received from the supplier during the report period.

From the PipeLine Summary Menu—

1. Click on the Shipment Order Report button.

PipeLine displays orders and associated data for the selected programs. By default, the Shipment Order report is set for all products and all shipment statuses with a 5-year reporting period.

Figure - PipeLine Summary Shipment Order Report screen



To limit the report's scope—

1. Click on the arrow next to the Select Supplier field, and select a supplier from the pull-down menu.
2. Click on the Select Status field, and select the shipment status you need.
3. Click on the From field, and type the beginning date of the report.
4. Click on the Through field, and type the ending date of the report.

The Page Breaks Between Status fields let you include page breaks between the shipment statuses listed in the report (when the report is generated for more than one shipment status).

1. Click on the Page Breaks Between Status field to include page breaks between shipment statuses.

Consumption Graph/Export

The Consumption graph groups consumption data by product ID or method ID.

Use this graph to compare relative shares and trends of distribution by product or method for multiple programs.

If you choose Method, the colors and legend will indicate what product and what program is represented by each Method ID bar. If you choose Method and Program Detail, you will see the relative shares of each program, but not product.

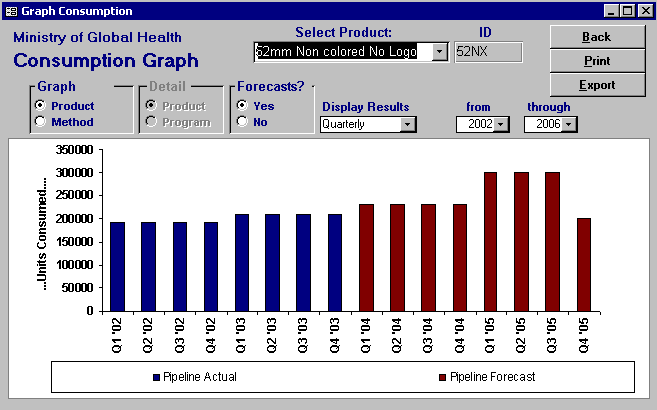
This is useful when programs use different, but comparable products, so the method distinction is more useful than the product distinction.

From the PipeLine Summary Menu—

1. Click on the Consumption Graph/Export button.

The graph is based on the last product/method and period updated or displayed. The Consumption Graph screen is displayed after the graph is created. Use this screen to create graphs for other products/methods.

Figure - PipeLine Summary Consumption Graph screen

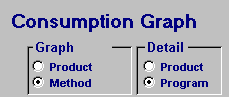


Graph Type

This section of the screen lets you determine the type of graph you will produce. Your options are Product or Method.

1. To create a graph by product, click on the arrow next to the Select Product field, and select a product from the pull-down menu.  
     
   Or, click on the Method field. Click on the arrow next to the Select Method field, and select a method from the pull-down menu to create a graph by method.

Figure - Graph options



When you create a consumption graph by method, the Detail section of the Consumption Graph screen becomes active. Product is selected by default, letting you create a graph for products associated with a selected method.

To create a graph for all products of a particular method through an aggregation of programs—

1. Click on the Program field.
2. Click on the arrow next to the Select Method field, and select a method from the pull-down menu.

Regardless of the type of graph you choose, select the remaining options.

By default, the graph includes consumption forecasts. To create a graph that excludes forecast data—

1. Click on the No field.

Figure - Forecast options

pipeline pics\sum-4.bmp

1. Click on the arrow next to the Display Results field, and select Monthly to display the data by month or Quarterly to display the data by quarter.

The default reporting period is five years (the previous two years, current year, and following two years). To change the default reporting period—

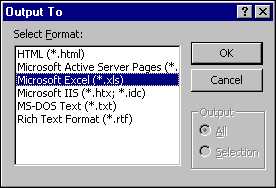
1. Click on the arrow next to the From field, and select the starting year of the report period.
2. Click on the arrow next to the Through field, and select the last year of the report period.

Exporting Summary Consumption Graph Data

Summary Consumption graph data can be exported for use with Microsoft Excel or other analysis software. After you create a consumption graph for a product or method—

1. Click on the Export button.

Figure - Output window



PipeLine displays the Output window (see Figure 8‑8).

1. Click on the file type for the export data.
2. Click on the OK button.

PipeLine opens another Output window, so you can name and save the export file.

1. Click on the OK button.  
     
   Or, click on the File Name field, and type the new export file name.
2. Click on the OK button to export and save the Consumption graph.
3. Click on the Back button to return to the PipeLine Summary menu.

Couple-Years of Protection (CYP) Graph

The CYP Graph converts consumption data by the CYP factor associated with a method.

Use this graph to measure and compare the extent of coverage provided by product or method across programs. The CYP factor is based on the percentage of one year where one unit of the method would provide a couple with contraception, if used properly.

You can compare coverage achievement over time between programs despite differences in method mix. A social marketing program may distribute more units and more volume and, perhaps, more value with methods like condoms and orals. But, a CYP analysis may indicate that a clinic NGO or MOH program may achieve more coverage because of the greater CYP factors of long-term methods such as Norplant, Depo-Provera, and IUDs.

Use this graph to evaluate and monitor relative program coverage or national coverage and achievement.

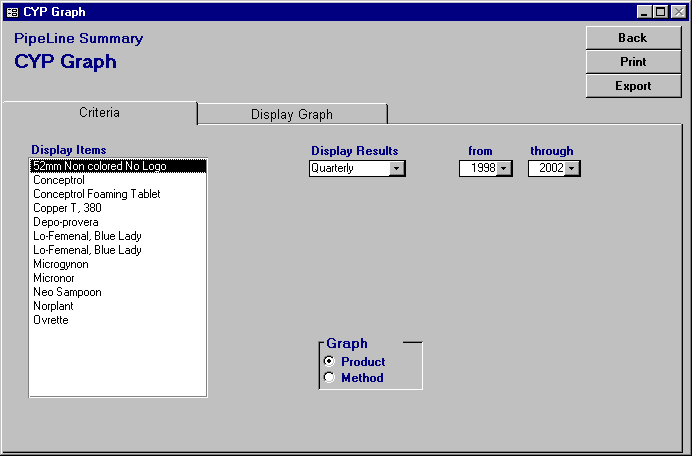
Creating the CYP Graph

Created the CYP graph is as follows.

From the PipeLine Summary Menu—

1. Click on the Couple-Years of Protection (CYP) Graph button.

Figure - PipeLine Summary CYP graph screen



The Display Items field shows the products or contraceptive methods that can be included in the graph.

Figure - Graph type selection

pipeline pics\cyp-pik.bmp

1. Click on the Method option to create a graph by method.  
     
   Or click on the Product option to create a graph by product.

After you set the graph for products or methods—

1. Click on the product (or method) you want to include.

To select multiple products (or methods)—

1. Hold down the Control key (<Ctrl>) and click on each product or method you want to include.
2. Click on the arrow next to the Display Results field, and select Monthly to display the data by month or Quarterly to display the data by quarter.
3. Click on the arrow next to the From field, and select the starting year of the report period from the pull-down menu.
4. Click on the arrow next to the Through field, and select the last year of the report period from the pull-down menu.

Displaying the Summary CYP Graph

Use the Display Graph tab to preview the CYP graph before printing it.

1. Click on the Display Graph tab.

Figure - CYP Graph display

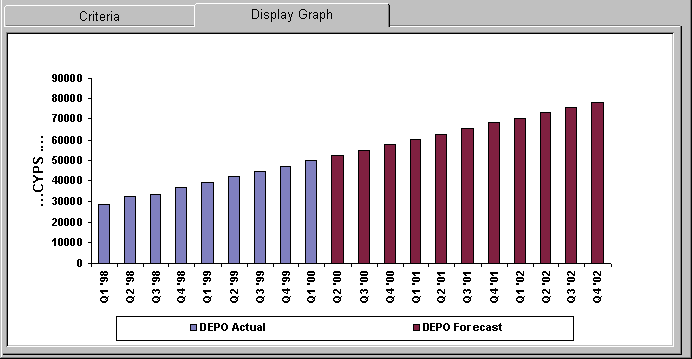


Figure — Export CYP Data window cyp-4a.bmp

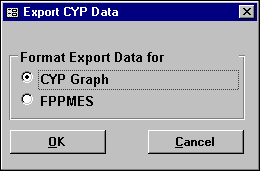
Exporting the CYP Graph

The Export button lets you export the data of a completed PipeLine Summary CYP graph. After you create a CYP graph, you can export the data as follows—

1. Click on the Export button.

PipeLine displays the Export CYP Data window. Use the options in this window to arrange the data in a Microsoft Office Excel spreadsheet.

Figure - Export CYP Data Window



CYP Graph

Incorporates the CYP factors into the consumption totals for ad hoc reports and graphs.

FPPMES

Arranges the data for the Family Planning Program Monitoring and Evaluation System.

The default is CYP Graph.

1. Click on the OK button to accept the default.  
     
   Or, click on FPPMES, and click on the OK button.

PipeLine opens another Output window, so you can name and save the export file. The default export file name is shown in the File Name field.

1. Click on the OK button to accept the default.  
     
   Or, click on the File Name field, and type the new export file name.
2. Click on the OK button to export, and save the Consumption graph.
3. Click on the Back button to return to the PipeLine Summary menu.

Previewing and Printing a Report or Graph

The procedure for printing a PipeLine Summary report or graph is the same regardless of the type you choose. To print a completed report or graph—

1. Click on the Print button to display the report or graph on your screen.
2. Click on the print icon  to send the report or graph to your default printer.
3. Click on the close icon  to return to the previous screen.
4. Click on the Back button to return to the PipeLine Summary menu.

**APPENDIX** **C: THE REDDICK VBA (RVBA) NAMING CONVENTIONS, VERSION 6.01**

*Copyright © 1992-1999 by Greg Reddick*

The purpose of the Reddick VBA (RVBA) Naming Conventions is to provide a guideline for naming objects in the Visual Basic for Applications (VBA) language. Having conventions is valuable in any programming project. When you use them, the name of the object conveys information about the meaning of the object. These conventions attempt to provide a way of standardizing that meaning across the body of VBA programmers.

VBA is implemented to interact with a host application-for example, Microsoft Access, Microsoft Visual Basic, AutoCAD, and Visio. The RVBA conventions cover all implementations of the VBA language, regardless of the host application. Some of the tags described in this document may not necessarily have an implementation within some of the particular host programs for VBA. The word object, in the context of this document, refers to simple variables and VBA objects, as well as to objects made available by the VBA host program.

While I am the editor of these conventions, they are the work of many people, including Charles Simonyi, who invented the Hungarian conventions on which these are based, and Stan Leszynski, who co-authored several versions of the conventions. Many others, too numerous to mention, have also contributed to the development and distribution of these conventions, but I’d especially like to thank Paul Litwin and Ken Getz who have made substantial contributions over the years.

These conventions are intended as a guideline. If you disagree with a particular part of the conventions, simply replace that part with what you think works better. However, keep in mind that future generations of programmers may need to understand those changes, and place a comment in the header of a module indicating what changes have been made. To be concise, the conventions are presented without rationalizations for how they were derived although each of the ideas presented has a considerable history to it.

Changes to the Conventions

Some of the tags in the version of the conventions presented here have changed from previous versions. Consider all previous tags to be grandfathered into the conventions--you don’t need to go back and make changes. For new development work, I leave it up to you to decide whether to use the older tags or the ones suggested here. In a few places in this document, older tags are shown in {braces}. As updates to this document are made, the current version can be found at the Xoc Software web site, [http://www.xoc.net.](http://www.xoc.net/)

An Introduction to Hungarian

The RVBA conventions are based on the Hungarian conventions for constructing object names, named for the native country of the inventor, Charles Simonyi. The objective of Hungarian is to convey information about the object concisely and efficiently. Hungarian takes some getting used to, but once adopted, it quickly becomes second nature. The format of a Hungarian object name is

[prefixes]tag[BaseName[Suffixes]]

The square brackets indicate optional parts of the object name. These components have the following meanings:

|  |  |
| --- | --- |
| Component | Meaning |
| Prefixes | Modify the tag to indicate additional information. Prefixes are all lowercase. They are usually picked from a standardized list of prefixes, given later in this document. |
| Tag | Short set of characters, usually mnemonic, that indicates the type of the object. The tag is all lowercase. It is usually selected from a standardized list of tags, given later in this document. |
| BaseName | One or more words that indicate what the object represents. Capitalize the first letter of each word in the BaseName. |
| Suffixes | Additional information about the meaning of the BaseName. Capitalize the first letter of each word in the Suffix. They are usually picked from a standardized list of suffixes, given later in this document. |

Notice that the only required part of the object name is the tag. This may seem counterintuitive; you may feel that the BaseName is the most important part of the object name. However, consider a generic procedure that operates on any form. The fact that the routine operates on a form is the important thing, not what that form represents. Because the routine may operate on forms of many different types, you do not necessarily need a BaseName. However, if you have more than one object of a type referenced in the routine, you must have a BaseName on all but one of the object names to differentiate them. In addition, unless the routine is generic, the BaseName conveys information about the variable. In most cases, a variable should include a BaseName.

Tags

Use the techniques described in the following sections to construct tags to indicate the data type of an object.

Variable tags

Use the tags listed in Table 21 for VBA data types. You can also use a specific tag instead of *obj* for any data type defined by the host application or one of its objects. (See the section “Host Application and Component Extensions to the Conventions” later in this document.)

Table - Tables for VBA Variables

| Tag | Object Type |
| --- | --- |
| bool {f, bln} | Boolean |
| byte {byt} | Byte |
| cur | Currency |
| date {dtm} | Date |
| dec | Decimal |
| dbl | Double |
| int | Integer |
| lng | Long |
| obj | Object |
| sng | Single |
| str | String |
| stf | String (fixed length) |
| var | Variant |

Here are several examples:

lngCount   
intValue   
strInput

You should explicitly declare all variables, each on a line by itself. Do not use the old-type declaration characters, such as %, &, and $. They are extraneous if you use the naming conventions, and there is no character for some of the data types, such as Boolean. You should always explicitly declare all variables of type Variant using the *As Variant* clause, even though it is the default in VBA. For example:

Dim intTotal As Integer   
Dim varField As Variant   
Dim strName As String

Constructing Properties Names

Properties of a class present a particular problem: should they include the naming convention to indicate the type? To be consistent with the rest of these naming conventions, they should. However, it is permitted to have property names without the tags, especially if the class is to be made available to customers who may not be familiar with these naming conventions.

Collection Tags

You treat a collection object with a special tag. You construct the tag using the data type of the collection followed by the letter s. For example, if you had a collection of Longs, the tag is lngs. If it was a collection of forms, the tag for the collection is frms. Although, in theory, a collection can hold objects of different data types, in practice, each of the data types in the collection is the same. If you do want to use different data types in a collection, use the objs tag. For example:

intsEntries   
frmsCustomerData   
objsMisc

Constants

Constants always have a data type in VBA. Because VBA will choose this data type for you if you don’t specify it, you should always specify the data type for a constant. Constants declared in the General Declarations section of a module should always have a scope keyword of Private or Public, and be prefixed by the scope prefixes *m* or *g*, respectively. A constant is indicated by appending the letter *c* to the end of the data type for the constant. For example:

Const intcGray As Integer = 3  
Private Const mdblcPi As Double = 3.14159265358979

Although this technique is the recommended method of naming constants, if you are more concerned about specifying that you are dealing with constants rather than their data type, you can alternatively use the generic tag *con* instead. For example:

Const conPi As Double = 3.14159265358979

Menu Items

The names of menu items should reflect their position in the menu hierarchy. All menu items should use the tag mnu, but the BaseName should indicate where in the hierarchy the menu item falls. Use *Sep* in the BaseName to indicate a menu separator bar, followed by an ordinal. For example:

mnuFile (on menu bar)  
mnuFileNew (on File popup menu)   
mnuFileNewForm (on File New flyout menu)   
mnuFileNewReport (on File New flyout menu)   
mnuFileSep1 (first separator bar on file popup menu)   
mnuFileSaveAs (on File popup menu)  
mnuFileSep2 (second separator bar on file popup menu)  
mnuFileExit (on File popup menu)  
mnuEdit (on menu bar)

Creating Data Types

VBA gives you three ways to create new data types: enumerated types, classes, and user-defined types. In each case, you will need to invent a new tag that represents the data type that you create.

Enumerated types

Groups of constants of the *long* data type should be made an enumerated type. Invent a tag for the type, append a “c,” and then define the enumerated constants using that tag. Because the name used in the Enum line is seen in the object browser, you can add a BaseName to the tag to spell out the abbreviation indicated by the tag. For example:

Public Enum ervcErrorValue   
ervcInvalidType = 205   
ervcValueOutOfBounds  
End Enum

The BaseName should be singular, so that the enumerated type should be ervcErrorValue, not ervcErrorValues. The tag that you invent for enumerated types can then be used for variables that can contain values of that type. For example:

Dim erv As ervcErrorValue  
Private Sub Example(ByVal ervCur As ervcErrorValue)

While VBA only provides enumerated types of groups of the long type, you can still create groups of constants of other types. Just create a set of constant definitions using an invented tag. For example:

Public Const estcError205 As String = "Invalid type"  
Public Const estcError206 As String = "Value out of bounds"

Unfortunately, because this technique doesn’t actually create a new type, you don’t get the benefit of the VBA compiler performing type checking for you. You create variables that will hold constants using a similar syntax to variables meant to hold instances of enumerated types. For example:

Dim estError As String

Tags for classes and user-defined types

A class defines a user-defined object. Because these invent a new data type, you will need to invent a new tag for the object. You can add a BaseName to the tag to spell out the abbreviation indicated by the tag. User-defined types are considered a simple class with only properties, but in all other ways are used the same as class modules. For example:

gphGlyph   
edtEdit  
Public Type grbGrabber

You then define variables to refer to instances of the class using the same tag: For example:

Dim gphNext As New gphGlyph   
Dim edtCurrent as edtEdit   
Dim grbHandle as grbGrabber

Polymorphism

In VBA, you use the Implements statement to derive classes from a base class. The tag for the derived class should use the same tag as the base class. The derived classes, though, should use a different BaseName from the base class. For example:

anmAnimal (base class)  
anmZebra (derived class of anmAnimal)  
anmElephant (derived class of anmAnimal)

This logic of naming derived classes is used with forms, which are all derived from the pre-defined Form base class and use the frm tag. If a variable is defined to be of the type of the base class, then use the tag, as usual. For example:

Dim anmArbitrary As anmAnimal  
Dim frmNew As Form

On the other hand, if you define a variable as an instance of a derived class, include the complete derived class name in the variable name. For example:

Dim anmZebraInstance As anmZebra  
Dim anmElephantExample As anmElephant  
Dim frmCustomerData As frmCustomer

Constructing Procedures

VBA procedures require you to name various items: procedure names, parameters, and labels. These objects are described in the following sections.

Constructing Procedure Names

VBA names event procedures, and you cannot change them. You should use the capitalization defined by the system. For user-defined procedure names, capitalize the first letter of each word in the name. For example:

cmdOK\_Click   
GetTitleBarString   
PerformInitialization

Procedures should always have a scope keyword, Public or Private, when they are declared. For example:

Public Function GetTitleBarString() As String  
Private Sub PerformInitialization

Naming Parameters

You should prefix all parameters in a procedure definition with ByVal or ByRef, even though ByRef is optional and redundant. Procedure parameters are named the same as simple variables of the same type, except that arguments passed by reference use the prefix “r.” For example:

Public Sub TestValue(ByVal intInput As Integer, ByRef rlngOutput As Long)  
Private Function GetReturnValue(ByVal strKey As String, \_  
ByRef rgph As Glyph) As Boolean

Naming Labels

Labels are named using upper and lower case, capitalizing the first letter of each word. For example:

ErrorHandler:  
ExitProcedure:

Prefixes

Prefixes modify an object tag to indicate more information about an object.

Arrays of Objects Prefix

Arrays of an object type use the prefix “a.” For example:

aintFontSizes   
astrNames

Index Prefix

You indicate an index into an array by the prefix i, and for consistency the data type should always be a long. You may also use the index prefix to index into other enumerated objects, such as a collection of user-defined classes. For example:

iaintFontSizes   
iastrNames   
igphsGlyphCollection

Prefixes for Scope and Lifetime

Three levels of scope exist for each variable in VBA: Public, Private, and Local. A variable also has a lifetime of the current procedure or the lifetime of the object in which it is defined. Use the prefixes in Table 22 to indicate scope and lifetime.

Table - Scope prefixes

|  |  |
| --- | --- |
| Prefix | Object Type |
| (none) | Local variable, procedure-level lifetime, declared with “Dim” |
| s | Local variable, object lifetime, declared with “Static” |
| m | Private (module) variable, object lifetime, declared with “Private” |
| g | Public (global) variable, object lifetime, declared with “Public” |

You also use the “m” and “g” constants with other objects, such as constants, to indicate their scope. For example:

intLocalVariable mintPrivateVariable gintPublicVariable mdblcPi

VBA allows several type declaration words for backward compatibility. The older keyword “Global” should always be replaced by “Public,” and the “Dim” keyword in the General Declarations section should be replaced by “Private.”

Other Prefixes

Table 23 lists and describes some other prefixes:

Table - Other commonly-used prefixes

|  |  |
| --- | --- |
| Prefix | Object Type |
| c | Count of some object type |
| h | Handle to a Windows object |
| r | Parameter passed by reference |

Here are some examples:

castrArray hWndForm

Suffixes

Suffixes modify the base name of an object, indicating additional information about a variable. You’ll likely create your own suffixes that are specific to your development work. Table 24 lists some generic VBA suffixes.

Table - Commonly-used suffixes

|  |  |
| --- | --- |
| Suffix | Object Type |
| Min | The absolute first element in an array or other kind of list |
| First | The first element to be used in an array or list during the current operation |
| Last | The last element to be used in an array or list during the current operation |
| Lim | The upper limit of elements to be used in an array or list. Lim is not a valid index. Generally, Lim equals Last + 1 |
| Max | The absolutely last element in an array or other kind of list |
| Cnt | Used with database elements to indicate that the item is a Counter. Counter fields are incremented by the system and are numbers of either type Long or type Replication Id. |

Here are some examples:

iastrNamesMin iastrNamesMax iaintFontSizesFirst igphsGlyphCollectionLast lngCustomerIdCnt varOrderIdCnt

File Names

When naming items stored on the disk, no tag is needed because the extension already gives the object type. For example:

Test.Frm (frmTest form)  
Globals.Bas (globals module)   
Glyph.Cls (gphGlyph class module)

Host Application and Component Extensions to the Conventions

Each host application for VBA, as well as each component that can be installed, has a set of objects it can use. This section defines tags for the objects in the various host applications and components.

Access 2000, Version 9.0 Objects

Table 25 lists Access object variable tags. Besides being used in code to refer to these object types, these same tags are used to name these kinds of objects in the form and report designers.

Table - Access object variable tags

| Tag | Object Type |
| --- | --- |
| aob | AccessObject |
| aops | AccessObjectProperties |
| aop | AccessObjectProperty |
| app | Application |
| bfr | BoundObjectFrame |
| chk | CheckBox |
| cbo | ComboBox |
| cmd | CommandButton |
| ctl | Control |
| ctls | Controls |
| ocx | CustomControl |
| dap | DataAccessPage |
| dcm | DoCmd |
| frm | Form |
| fcd | FormatCondition |
| fcds | FormatConditions |
| frms | Forms |
| grl | GroupLevel |
| hyp | Hyperlink |
| img | Image |
| lbl | Label |
| lin | Line |
| lst | ListBox |
| bas | Module |
| ole | ObjectFrame |
| opt | OptionButton |
| fra | OptionGroup (frame) |
| brk | PageBreak |
| pal | PaletteButton |
| prps | Properties |
| shp | Rectangle |
| ref | Reference |
| refs | References |
| rpt | Report |
| rpts | Reports |
| scr | Screen |
| sec | Section |
| sfr | SubForm |
| srp | SubReport |
| tab | TabControl |
| txt | TextBox |
| tgl | ToggleButton |

Some examples:

txtName lblInput

For ActiveX custom controls, you can use the tag ocx as specified in Table 25 or more specific object tags that are listed later in this document in Tables 34 and 35. For an ActiveX control that doesn't appear in the Tables 34 or 35, you can either use ocx or invent a new tag.

DAO 3.6 Objects

DAO is the programmatic interface to the Jet database engine shared by Access, Visual Basic, and Visual C++. The tags for DAO 3.6 objects are shown in Table 26.

Table - DAO object tags

| Tag | Object Type |
| --- | --- |
| cnt | Container |
| cnts | Containers |
| db | Database |
| dbs | Databases |
| dbe | DBEngine |
| doc | Document |
| docs | Documents |
| err | Error |
| errs | Errors |
| fld | Field |
| flds | Fields |
| grp | Group |
| grps | Groups |
| idx | Index |
| idxs | Indexes |
| prm | Parameter |
| prms | Parameters |
| pdbe | PrivDBEngine |
| prp | Property |
| prps | Properties |
| qry | QueryDef |
| qrys | QueryDefs |
| rst | Recordset |
| rsts | Recordsets |
| rel | Relation |
| rels | Relations |
| tbl | TableDef |
| tbls | TableDefs |
| usr | User |
| usrs | Users |
| wrk | Workspace |
| wrks | Workspaces |

Here are some examples:

rstCustomers idxPrimaryKey

Table 27 lists the tags used to identify types of objects in a database.

Table - Access Database Explorer object tags

| Tag | Object Type |
| --- | --- |
| tbl | Table |
| qry | Query |
| frm | Form |
| rpt | Report |
| mcr | Macro |
| bas | Module |
| dap | DataAccessPage |

If you wish, you can use tags that are more exact or suffixes to identify the purpose and type of a database object. If you use the suffix, use the tag given from Table 27 to indicate the type. Use either the tag or the suffix found along with the more general tag, but not both. The tags and suffixes are shown in Table 28.

Table - Specific object tags and suffixes for Access Database Explorer objects

| Tag | Suffix | Object Type |
| --- | --- | --- |
| tlkp | Lookup | Table (lookup) |
| qsel | (none) | Query (select) |
| qapp | Append | Query (append) |
| qxtb | XTab | Query (crosstab) |
| qddl | DDL | Query (DDL) |
| qdel | Delete | Query (delete) |
| qflt | Filter | Query (filter) |
| qlkp | Lookup | Query (lookup) |
| qmak | MakeTable | Query (make table) |
| qspt | PassThru | Query (SQL pass-through) |
| qtot | Totals | Query (totals) |
| quni | Union | Query (union) |
| qupd | Update | Query (update) |
| fdlg | Dlg | Form (dialog) |
| fmnu | Mnu | Form (menu) |
| fmsg | Msg | Form (message) |
| fsfr | SubForm | Form (subform) |
| rsrp | SubReport | Form (subreport) |
| mmnu | Mnu | Macro (menu) |

Here are some examples:

tblValidNamesLookup tlkpValidNames fmsgError mmnuFileMnu

When naming objects in a database, do not use spaces. Instead, capitalize the first letter of each word. For example, instead of Quarterly Sales Values Table, use tblQuarterlySalesValues.

There is strong debate over whether fields in a table should have tags. Whether you use them is up to you. However, if you do use them, use the tags from Table 29.

Table - Field tags (if you decide to use them)

| Tag | Object Type |
| --- | --- |
| lng | Autoincrementing (either sequential or random) Long (used with the suffix Cnt) |
| bin | Binary |
| byte | Byte |
| cur | Currency |
| date | Date/time |
| dbl | Double |
| guid | Globally unique identified (GUID) used for replication AutoIncrement fields |
| int | Integer |
| lng | Long |
| mem | Memo |
| ole | OLE |
| sng | Single |
| str | Text |
| bool | Yes/No |

Visual Basic 6.0 Objects

Table 30 shows the tags for Visual Basic 6.0 objects.

Table - Visual Basic 6.0 object tags

| Tag | Object Type |
| --- | --- |
| app | App |
| chk | CheckBox |
| clp | Clipboard |
| cbo | ComboBox |
| cmd | CommandButton |
| ctl | Control |
| dat | Data |
| dir | DirListBox |
| drv | DriveListBox |
| fil | FileListBox |
| frm | Form |
| fra | Frame |
| glb | Global |
| hsb | HScrollBar |
| img | Image |
| lbl | Label |
| lics | Licenses |
| lin | Line |
| lst | ListBox |
| mdi | MDIForm |
| mnu | Menu |
| ole | OLE |
| opt | OptionButton |
| pic | PictureBox |
| prt | Printer |
| prp | PropertyPage |
| scr | Screen |
| shp | Shape |
| txt | TextBox |
| tmr | Timer |
| uctl | UserControl |
| udoc | UserDocument |
| vsb | VscrollBar |

Microsoft ActiveX Data Objects 2.1 Tags

Office 2000 provides version 2.1 of the ActiveX Data Objects library. Table 31 lists the recommended tags for this version of ADO.

Note: Many of the ADO, ADOX, and JRO tags overlap with existing DAO tags. Make sure you include the object library name in all references in your code, so there’s never any possibility of confusion. For example, use

Dim rst As ADODB.Recordset

or

Dim cat As ADOX.Catalog

rather than using the object types without the library name. This will not only make your code more explicit and avoid confusion about the source of the object, but will also make your code run a bit faster.

Table - ADO 2.1 Object tags

| Tag | Object Type |
| --- | --- |
| cmn {cmd} | Command |
| cnn {cnx} | Connection |
| err | Error |
| errs | Errors |
| fld | Field |
| flds | Fields |
| prm | Parameter |
| prms | Parameters |
| prps | Properties |
| prp | Property |
| rst | Recordset |

Microsoft ADO Ext. 2.1 for DDL and Security (ADOX) Tags

In order to support DDL and security objects within Jet database, Microsoft provides ADOX, an additional

ADO library of objects. Table 32 lists tags for the ADOX objects.

Table - ADOX Object tags

| Tag | Object Type |
| --- | --- |
| cat | Catalog |
| clms | Column |
| clm | Columns |
| cmd | Command |
| grp | Group |
| grps | Groups |
| idx | Index |
| idxs | Indexes |
| key | Key |
| keys | Keys |
| prc | Procedure |
| prcs | Procedures |
| prps | Properties |
| prp | Property |
| tbl | Table |
| tbls | Tables |
| usr | User |
| usrs | Users |
| vw | View |
| vws | Views |

Microsoft Jet and Replication Objects 2.1

In order to support Jet’s replication features, ADO provides another library (JRO). Table 33 lists suggested tags for the JRO objects.

Table - JRO object tags

| Tag | Object Type |
| --- | --- |
| flt | Filter |
| flts | Filters |
| jet | JetEngine |
| rpl | Replica |

Microsoft SQL Server and Microsoft Data Engine (MSDE) Objects

Table 34 lists RVBA tags for Microsoft SQL Server and the Microsoft Data Engine (a limited-connection version of SQL Server 7) objects.

Table - SQL Server/MSDE object tags

| Tag | Object Type |
| --- | --- |
| tbl | table |
| proc | stored procedure |
| trg | trigger |
| qry | view |
| dgm | database diagram |
| pk | primary key |
| fk | foreign key |
| idx | other (non-key) index |
| rul | check constraint |
| def | default |

Microsoft Common Control Objects

Windows 95 and Windows NT have a set of common controls that are accessible from VBA. Table 35 lists the tags for objects created using these controls.

Table - Microsoft Common Control Object tags

| Tag | Object Type |
| --- | --- |
| ani | Animation |
| btn | Button (Toolbar) |
| bmn | ButtonMenu (Toolbar) |
| bmns | ButtonMenus (Toolbar) |
| bnd | Band (CoolBar) |
| bnds | Bands (CoolBar) |
| bnp | BandsPage (CoolBar) |
| btns | Buttons (Toolbar) |
| cbr | CoolBar |
| cbp | CoolBarPage (CoolBar) |
| hdr | ColumnHeader (ListView) |
| hdrs | ColumnHeaders (ListView) |
| cbi | ComboItem (ImageCombo) |
| cbis | ComboItems (ImageCombo) |
| ctls | Controls |
| dto | DataObject |
| dtf | DataObjectFiles |
| dtp | DTPicker |
| fsb | FlatScrollBar |
| imc | ImageCombo |
| iml | ImageList |
| lim | ListImage |
| lims | ListImages |
| lit | ListItem (ListView) |
| lits | ListItems (ListView) |
| lsi | ListSubItem (ListView) |
| lsis | ListSubItems (ListView) |
| lvw | ListView |
| mvw | MonthView |
| nod | Node (TreeView) |
| nods | Nodes (TreeView) |
| pnl | Panel (Status Bar) |
| pnls | Panels (Status Bar) |
| prb | ProgressBar |
| sld | Slider |
| sbr | StatusBar |
| tab | Tab (Tab Strip) |
| tabs | Tabs (Tab Strip) |
| tbs | TabStrip |
| tbr | Toolbar |
| tvw | TreeView |
| udn | UpDown |

Other Custom Controls and Objects

Finally, Table 36 lists the tags for other commonly used custom controls and objects.

Table - Tags for commonly-used custom controls

| Tag | Object Type |
| --- | --- |
| cdl | CommonDialog (Common Dialog) |
| dbc | DBCombo (Data Bound Combo Box) |
| dbg | DBGrid (Data Bound Grid) |
| dls | DBList (Data Bound List Box) |
| gau | Gauge (Gauge) |
| gph | Graph (Graph) |
| grd | Grid (Grid) |
| msg | MAPIMessages (Messaging API Message Control) |
| ses | MAPISession (Messaging API Session Control) |
| msk | MaskEdBox (Masked Edit Textbox) |
| key | MhState (Key State) |
| mmc | MMControl (Multimedia Control) |
| com | MSComm (Communication Port) |
| out | Outline (Outline Control) |
| pcl | PictureClip (Picture Clip Control) |
| rtf | RichTextBox (Rich Textbox) |
| spn | SpinButton (Spin Button) |

Summary

Using a naming convention requires a considerable initial effort on your part. The payoff comes when either you or another programmer has to revisit your code later. Using the conventions given here will make your code more readable and maintainable.

*Greg Reddick is the President of Xoc Software, a software development company developing programs in Visual Basic, Microsoft Access, C/C++, and for the web. He leads training seminars in Visual Basic for Application Developers Training Company (AppDev). In a previous life, he worked for four years on the Access development team at Microsoft. Greg can be reached at* [*mailto:grr@xoc.net*](mailto:mailto:grr@xoc.net) *or from the Xoc Software web site,* [*http://www.xoc.net.*](http://www.xoc.net/)

APPENDIX D: THE REDDICK VBA (RVBA) CODING CONVENTIONS (VERSION 0.90)

*Copyright © 1999 by Greg Reddick*

What follows are the Reddick VBA (RVBA) Coding Conventions. The objectives of the conventions are to make code:

More readable: Conventions allow a reader to understand the meaning of the code with less effort.

* More maintainable: The code can be more reliably changed to fix bugs and enhance functionality.
* More reliable: The code is more likely to perform as expected.
* More efficient: The code performs faster or consumes fewer resources.

These conventions are separate from the RVBA Naming Conventions and may be adopted without adopting the naming conventions. The current version of these conventions can always be found on the Xoc Software web site: http://www.xoc.net.

More rational is provided for these recommendations than is given in the RVBA Naming Conventions. In most cases, there are good rationales for the given conventions. However, in some cases an arbitrary decision was made to select one convention from a set of reasonable alternatives. The other reasonable alternatives to the conventions placed in {braces} at the end of a section. In some cases, a topic only relates to the Visual Basic 6.0 development environment, as opposed to VBA in general. In those cases, the topic is marked with [VB6] after the topic heading.

No set of conventions can cover every case or every consideration. The general rule is that exceptions to the conventions can be made with the approval of the programming team after careful consideration.

The sections are listed in alphabetical order to facilitate their use as a reference work. However, this makes the flow of the document unusual for casual reading as some topics are much more technical than others.

Arrays

Always specify the both the lower and upper bound of an array. This makes explicit whether element zero of the array is a valid element or not. For example:

Dim astrValue(1 To 10) As String

By convention the index variable used to walk an array should always be a Long data type. This assures that if the array size grows past 32767 elements when maintaining the program that the index variable can still address all elements in the array

When walking an array, always use the VBA LBound and UBound functions to visit each item. This makes sure that every item in the array is visited. For example:

Dim iastrValue As Long  
 For iastrValue = LBound(astrValue) To UBound(astrValue)  
 MsgBox astrValue(iastrValue)  
 Next iastrValue

Assertions

VBA provides a built-in assertion mechanism through Debug.Assert. If the expression following the Debug.Assert evaluates to True, the code continues. If the expression evaluates to False, VBA enters Break mode as if a breakpoint had been set on that line. The line shown here acts as a hard coded breakpoint:

Debug.Assert False

Assertions that do not include a function call in the expression are removed by the compiler when an executable is made, so they only apply to debugging inside the VBA environment. Assertions with a function call in the expression will remain in the executable, but the resulting value of the expression is discarded. VBA doesn't remove function calls because they may have side effects, but discards the return value from the function.

Any time that there is an assumption in the code about the state of the program, there should be an assertion that states the assumption. For example, suppose that a procedure includes this code:

Select Case intValue  
 Case 1  
 MsgBox "Aircraft"  
 Case 2  
 MsgBox "AutoMobile"  
 Case 3  
 MsgBox "SnowMobile"  
 End Select

This code assumes that the value of intValue is between one and three. However, if through some bug, intValue had the value of zero or four, this code doesn't work right. The result is that no MsgBox appears at all. Tracking down why the MsgBox doesn't appear is time consuming. Instead, the code could be written one of two other ways. Either:

Debug.Assert intValue >= 1 And intValue <= 3  
 Select Case intValue  
 Case 1  
 MsgBox "Aircraft"  
 Case 2  
 MsgBox "AutoMobile"  
 Case 3  
 MsgBox "SnowMobile"  
 End Select

Or

Select Case intValue  
 Case 1  
 MsgBox "Aircraft"  
 Case 2  
 MsgBox "AutoMobile"  
 Case 3  
 MsgBox "SnowMobile"  
 Case Else  
 Debug.Assert False  
 End Select

In general, every Select/End Select block should have a Case Else to trap unexpected values. If the Case Else should never occur, then a Debug.Assert False should be inserted into the block. If the code is correctly written to handle 1 To 3, but zero and four are allowed values, the code should be written with a comment in the Case Else block to indicate that this is expected, like this:

Select Case intValue  
 Case 1  
 MsgBox "Aircraft"  
 Case 2  
 MsgBox "AutoMobile"  
 Case 3   
 MsgBox "SnowMobile"   
Case Else 'Do nothing   
End Select

Assertions trap logic errors early. Rather than waiting to see the results of a bug in the use interface, there is immediate feedback that the bug has occurred. Assertions are only effective if they are present, which means that they have to be added when writing the code. Any logic error that is fixed in the code is a good indication that some additional assertions need to be added.

Comments

A comment in VBA starts with an apostrophe and ends at the end of the line. Comments may be placed on a line by themselves or at the end of a line. A comment starts with the apostrophe followed immediately by the text with no space between the two.

The comment at the end of a line should be used in only a few places:

* At the end of a declaration line
* On a Case line
* On the line that ends a block to indicate what block is being ended. For example on a set of nested If/End If blocks, a comment on the End If line may say what If block is completed. This is especially useful if the block spans several screens.

Examples:

Dim dateUTC As String 'time in Univeral Coordinated Time Case 11 'Division by Zero

If the end of a line comment line exceeds the 80 characters line limit, continue the comment on the next line indented by one tab stop. For example:

Case 35602 'This key is already associated with an element of this 'collection Set nodChild = tvw.Nodes.Item(cci.Guid)

All other comments should be placed on a separate line above the line they are documenting and indented to the same level. A comment of this sort is generally preceded by a blank line unless it is the first line of an indented block. For example:

vt = vti.VarType

'Special hack for analyzing my code If LCase$(Left$(strParamName, Len(strcDecPrefix))) = strcDecPrefix Then strDataType = strDecimal End If

If it is the first line of an indented block, it is not preceded by a blank line. For example:

If mboolShowProperties Then 'Show properties for each member For Each mi In ci.Members

Comments should state the intention of the code not how it performs the task. This is an example of a worthless comment:

'Place the VarType into the vt variable vt = vti.VarType

It is worse than no comment at all. The comment is wrong if the code changes to use the variable name vtCur instead of vt without changing the comment. When reading a comment that doesn't match the code, the question becomes whether the comment is correct or the code is correct. Usually it is the comment that is wrong, but it may take some time to prove that. A wrong comment can be worse than no comment at all. A comment that says the same thing as the following line of code is worthless. In general, don't write comments that have to be maintained, because in the real-world comments frequently aren't maintained.

A comment that states the intention of the code, though, may be useful. For example:

'Store VarType for recovery in error condition. vt = vti.VarType

However, use these comments only when the intention is not immediately clear when reading the code. Instead strive to make the code self explanatory, through good naming and coding conventions.

Constants

Always give constants an explicit data type. For example:

Private Const dblcPi As Double = 3.14159265358979

If a literal value other than zero or one appears in the code, consideration should be given to whether it makes things more readable and maintainable to replace it with a constant. Replace a magic number used more than once in the code with a constant.

Global constants are allowed and encouraged. Replace sets of constants of the data type long with enumerated types using Enum.

Date Functions and Date Variables

Be careful about using the VBA date functions: Date, DateAdd, DateSerial, DateValue, and Now. These functions return a variant containing a date. If implicit type conversion to turns the return value into a string, the string representation of the date displays a two-digit year number. That year number is, of course, not Y2K compliant. This also applies to allowing a variable of type Date to be converted into a string. Instead, use the Format$ function to convert the date into a string. For example:

strValue = Format$(Date, "mm/dd/yyyy")

Default Properties

Using default properties makes code difficult to read. VBA allows you to just use the name of a textbox and looks up the default property, Text. For example:

MsgBox txtValue

This prints the value of the txtValue textbox. On the other hand, it is much clearer to say:

MsgBox txtValue.Text

To even be more explicit, it could even be expressed as:

MsgBox Me.txtValue.Text

This, however, does not add any additional worth because all references to a control in a module from a form are implicitly on Me.

The reason to be explicit about default properties is to keep the programmer from having to figure out what property is being referenced. This is especially true when referencing ActiveX controls and ActiveX DLLs where the default properties are obscure. For example, when an ADO field is referenced, you are allowed to say:

varValue = rst!strFirstName

This references the Value property of the strFirstName field. However, it is much clearer to say:

varValue = rst.Fields.Item("strFirstName").Value

This code doesn't use any default properties and retrieves the same value.

Deprecated Features

Avoid using features Visual Basic supports only for backwards compatibility. Avoid using undocumented features. Also, avoid using functionality that VBA has replaced with functionality that is more modern. Some examples of these kinds of features:

* %, &, $, Etc. in declaration of variables
* Rem statements
* Line numbers (except in conjunction with the Erl function in special error handling situations)
* Single line If statements (use If/End If blocks instead). For example, don't use:

If boolValue Then MsgBox "Hi There"

* While/Wend loops (replace with Do While/Loop)
* Variables declared with Global (use Public instead). Using Dim in the General Declarations section (use Private instead)
* Using user defined types except in the case of Windows API calls or reading fixed width record files (use Class modules instead)
* Gosub
* The End statement in most cases (simply unload the last form in a standard EXE instead)

Disambiguation

When referencing classes from an ActiveX library, always use the library name to explicitly tell VBA from what library to get the class. If you don't, then VBA will use the order of the libraries in the References dialog to determine from which library it gets the class. The library name always appears in the upper left-hand listbox of the VBA object browser. For example, if there are references to both the Access and Excel object libraries, then this is ambiguous:

Dim appObj As Application

Because both the Access and Excel libraries include an Application class, which Application class is referenced depends on which one appears first in the References dialog. Instead, it should be declared like this:

Dim appObj As Excel.Application

Microsoft refers to this as "disambiguation". With this declaration, it does not matter what the order of the libraries is inside the References dialog, as appObj will always refer to the Excel Application object. All references to class names in libraries should include the disambiguating library name.

DLL Base Address [VB6]

The base address is the location that the DLL is loaded into memory. If two DLLs are loaded into the same base address, then VBA moves the second DLL to a new address. VBA then has to modify the binary code within the DLL's address space to reflect the new address. This slows down loading the second DLL.

Libraries used together should start at different base addresses. In Visual Basic, enter the Base Address for a library in the Project Properties dialog Compile tab. Enter a random number base address different than any other used at the same time.

Dollar Sign ($) Functions

If the result of a function is used as a string or assigned it to a string variable, use the $ form of the function. This results in faster executing code, because a conversion from a variant to a string is unnecessary. For example, this is proper usage of dollar sign functions:

If LCase$(Left$(strParamName, Len(strcDecPrefix))) = strcDecPrefix Then

This example calls the LCase$ and Left$ functions instead of the LCase and Left functions because the result is used as a String. If the result is used as a Variant, then call the LCase and Left functions instead.

The $ version of the function returns the same value as the Variant version. The one except to the rule is the VBA Date function. The Date function should always be used because the Date$ function doesn't behave correctly. The Date$ always returns information in mm-dd-yyyy format regardless of the Windows localization settings, whereas the Date function uses the localization settings.

Error Handling

A procedure should always include runtime error handling. In general, Error handling should be blocked out the same way in every procedure, as shown in this example:

Private Sub Test() On Error GoTo ErrorHandler

'Code for the procedure goes here

ExitProcedure:

On Error Resume Next

'Cleanup code for the procedure goes here Exit Sub ErrorHandler:

Select Case Err.Number

'Case statements for expected errors goes here

Case Else

Call UnexpectedError(Err.Number, Err.Description, Err.Source, \_

Err.HelpFile, Err.HelpContext)

End Select

Resume ExitProcedure End Sub

Use the label names shown in the example, although the label names have been arbitrarily chosen. Notice that the Exit Sub and ErrorHandler label are left justified making them easily findable. Case statements for expected errors should be given with the error number and a comment with the error message. For example:

Select Case Err.Number

'Case statements for expected errors go here

Case 11 'Division by zero

MsgBox "Zero isn't a valid divisor", vbExclamation, Me.Caption

Case Else

The UnexpectedError routine is a global routine that is only called in a condition where a runtime error that isn’t expected is received, so that there is a bug in the problem. This procedure should log the error message. At the absolute minimum it should just look like this, but ideally it should do a lot more to log the error:

Public Sub UnexpectedError(ByVal lngNumber As Long, \_

ByVal strDescription As String, ByVal strSource As String, \_ ByVal strHelpfile As String, ByVal lngHelpContext As Long) On Error Resume Next MsgBox "[" & strSource & "]" & vbCrLf & "Run-time error '" \_

& CStr(lngNumber) & "':" \_ & vbCrLf & vbCrLf & strDescription, vbExclamation, App.Title, \_ strHelpfile, lngHelpContext

Debug.Print "Case " & CStr(lngNumber) & " '" & strDescription Debug.Assert False End Sub

The first executable line of every procedure should be the On Error GoTo ErrorHandler line. The only exception to the rule is when a procedure checks the values of its arguments and generates a runtime error when they are invalid. In this case, the checking code comes before the On Error GoTo line. For example

Public Sub Test(ByVal intValue As Integer) If intValue < 1 Or intValue > 10 Then Call Err.Raise(Number:=lngcInvalidValue, \_

Description:=strcInvalidValue) End If On Error Goto ErrorHandler

Exiting a Procedure

In general, a procedure should only have one exit point. Having one exit point makes it easier to read the code and understand when and where it exits. If you use the code mentioned in the Error Handling code section, that exit point is the Exit Sub, Exit Function, or Exit Property line at the top of the error handling. The only other way to exit the procedure should be through using Err.Raise. These Err.Raise lines should occur either before the On Error GoTo line when validating the parameters (see Error Handling) or inside the error handler.

In a few cases, there may be a need to raise an error inside the body of the procedure. In such cases, you should explicitly set any object variables to Nothing (see Nothing), and then exit the procedure. In such cases, the exiting the procedure should be explicitly detailed by a comment that shows the exit, consisting of an arrow stretching to 80 character right margin. The On Error GoTo 0 statement has to be used to turn off error handling for this procedure before executing the Err.Raise. For example, if this code appears somewhere after the On Error GoTo line, it should be written like this to make it explicit that there is an exit point in the middle of the procedure:

If intValue > 1000 Then 'Raise an Error-------------------------------------------------------> Set rst = Nothing On Error Goto 0 Call Err.Raise(Number:=lngcInvalidValue, \_

Description:=strcInvalidValue) End if

For/Next and For Each/Next Loops

The index variable used in the For/Next loop should be specified on the Next line. This makes it explicit which For loop is being completed. For example:

Dim iastrValue As Long For iastrValue = LBound(astrValue) To UBound(astrValue) MsgBox astrValue(iastrValue) Next iastrValue

The object variable used to walk the collection should be placed on the Next line in a For Each/Next loop. For example:

Dim frm As Form For Each frm in Forms

If Not (frm Is Me) Then Unload Me End If Next frm

GoTo Statements

You can usually avoid using GoTo statements in VBA code. Use GoTo statements only when the alternative code is not as clear as the GoTo statement. A common reason to use a GoTo statement is to jump out of nested loops. For Example:

For iastrOuterLoop = 1 To 10

For iastrInnerLoop = 1 To 100 'some other code If astr(iastrOuterLoop, iastrInnerLoop) = "Done" Then

GoTo ExitNestedLoops End If 'some other code

Next iastrInnerLoop Next iastrOuterLoop ExitNestedLoops: 'More code here

Headers

Each module should start with a header code that looks something like this:

'$Header: $ '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Option Explicit 'This module includes definitions of Windows API calls

The line of asterisks is an apostrophe followed by 79 asterisks. See the section on Long Lines.

Each Public procedure should begin with a header block that looks something like this:

Public Sub Almanac(ByVal lngTrecena As Long, ByVal vein As veinc, \_ ByVal lngRows As Long, ByRef alngBlack() As Long, \_ ByRef alngRed() As Long, ByRef aveinRowStart() As veinc, \_ ByRef aveinAlmanac() As veinc, ByRef lngComplete As Long) 'Generates a Maya almanac

'If lngComplete returns zero then it is an almanac, if it is non-zero, 'then it misses completing and you'll need to report that. You will still 'need to handle the error encNotAnAlmanac because the black numbers 'in alngBlack must wrap back to the starting lngTrecena. On Error GoTo ErrorHandler

Read/write values allowed are [in], [out], and [inout].

Event procedures do not need a header unless the scope is changed to Public. Private procedures may need the header depending on the context. Note that the name of the routine is not referenced in the comments, making it possible to change the name of the procedure without changing the comments. No change history or coding history is included. Histories should be maintained by source code control systems, not by programmers since they are rarely properly kept up to date.

|  |  |
| --- | --- |
| 'lngTrecena [in] | Upper left corner trecena |
| 'vein [in] | Upper left corner veintena |
| 'lngRows [in] | Number of rows in the almanac |
| 'alngBlack() [in] | Black distance numbers across almanac |
| 'alngRed() [out] | Calculated Red trecena numbers across almanac |
| 'aveinRowStart() [out] | Calculated Leftmost shown veintenas in almanac |
| 'aveinAlmanac() [out] | Actual veintenas implied by almanac |
| 'lngComplete [out] | Number almanac misses completing by. |
| 'Return value: | None |

The comments are addressed to the person calling the procedure, and should include just enough information to tell the person how to call the procedure and use the returned values. After the On Error GoTo, other comments can be placed describing algorithms and other implementation details, if needed (although see the section on Comments).

Indenting

Tab stops should be set at four spaces. No member of a programming team should vary this number, as it makes editing other members of the team's code difficult.

All code inside a block should be indented one tab stop from the surrounding code, with exceptions noted elsewhere in this document. Indenting blocks makes finding the start and end of the block easy. A block is defined as the code that falls between the following keywords:

* Do/Loop
* Enum/End Enum
* For/Next
* For Each/Next
* Function/Exit Function/End Function
* If/Else/ElseIf/End If
* #If/#Else/#ElseIf/#End If
* Property/Exit Property/End Property
* Sub/Exit Sub/End Sub
* Type/End Type
* With/End With

For example:

For Each ci In tlio.Constants

Set nodChild = tvw.Nodes.Add(Relative:=nod.Key, \_ Relationship:=tvwChild, \_ Key:=ci.Guid & ci.Name, Text:=ci.Name, Image:=strcEnum)

nodChild.EnsureVisible DoEvents If mboolShowProperties Then

For Each mi In ci.Members

Set nodEnumChild = tvw.Nodes.Add(Relative:=nodChild.Key, \_ Relationship:=tvwChild, Text:=mi.Name & strcEquals & \_ mi.Value, Image:=strcConstant)

nodEnumChild.EnsureVisible DoEvents Next mi End If Next ci

See also the section on Select/End Select Blocks.

{Alternative: The entire programming team may standardize on another number of spaces.}

{Alternative: Exit Function, Exit Property, and Exit Sub statements may be indented to the level of the surrounding code.}

Instantiation

An object variable should not be declared with New on the line it is declared on, unless there is a good reason to do so. The declaration should instead be broken into two lines. For example:

Dim rst As ADODB.Recordset Set rst = New ADODB.Recordset

Not this:

Dim rst As New ADODB.Recordset

Breaking it into two lines causes each reference to the rst variable to execute slightly faster. In addition, the object variable can be tested to see if it contains the value Nothing. For example:

If rst Is Nothing Then MsgBox "rst not initialized" End If

If a one-line declaration is used, the above code would never execute the MsgBox because the reference to the rst variable in the If statement causes the object to be instantiated before the Is operator is evaluated. For Private and Public object variables, occasionally the convenience of using the New keyword outweighs the performance benefit, so the one-line declaration may still be used.

Labels

Labels in the code should be left justified, regardless of the indenting level of the surrounding code. They should appear on a line by themselves. For example:

ExitProcedure: On Error Resume Next

Long Lines of Code

VBA code editors will scroll a line of code to make the end visible. However, this makes it difficult to read the code quickly. It also means the code is not understandable if placed into a media that doesn't scroll, such as a paper print out or a book. For these reasons, the length of lines should be restricted.

A physical line of code should not exceed 80 characters. If a logical line of code exceeds 80 characters, then the line should be broken into two or more physical lines using the underscore line continuation character. All physical lines in the logical line following the first physical line should be indented one tab stop (four spaces) from the first physical line.

It may help to place a line at the top of the module with an apostrophe followed by 79 asterisks. Then the code window of the VBA editor can be sized to barely make the last asterisk visible. A fixed width font, such as Courier New, should be used to display the code in the VBA code window.

You should choose an appropriate place to break the line to enhance the maximum readability of the remaining code. When breaking lines that have a list separated by commas, you should break the line after a comma and before the next non-space character. For example:

Private Sub GetFiles(ByRef fso As Scripting.FileSystemObject, \_ ByRef fld As Scripting.Folder)

When breaking a line that is an expression built by operators, break the line before an operator of the expression. For example when the expression is built of string concatenation operators, break it like this:

strParameters = strParameters & strAdd \_ & strPassingConvention & pmi.Name & strArray \_ & strcAs & strDataType & strDefault

The next line becomes more readable this way.

If you have a long literal string, you may have to break the line like this:

strValue = "This is a very, very long string that will cause the code " \_ & "to wrap. Because of this, you will need to break it."

In such cases, break it before the start of a word. Note that VBA performs the string concatenation at runtime, so this has performance considerations. In many cases, the string should be placed into a constant, an entry in resource file, or a database field and retrieved from there.

Comments should never be continued. When a comment exceeds 80 characters, continue the comment on the next line preceded by another apostrophe. See the section on comments.

Don't overly indent lines. Move overly indented code to a new procedure and call it from the original. In general, code should not need to be indented more than eight tab stops.

{Alternative: Place operators at the end of the line before the line continuation character instead of on the next line.}

Nothing

Explicitly set Object variables to Nothing before allowing the variable to be destroyed. This is especially true of object variables declared with the Dim keyword. For Example:

Public Sub Test(ByVal intValue As Integer)

'error handling omitted for clarity Dim rst As ADODB.RecordSet Set rst = New ADODB.RecordSet 'More code here Set rst = Nothing

End Sub

Setting the object variable to Nothing is not just good programming practice. If the rst object has code in its Class\_Terminate event handler, that code can mess with global variables and objects.

In addition, set the object variable to Nothing is before exiting the procedure with Err.Raise. For example:

Private Sub Test(ByVal intValue As Integer)

'error handling omitted for clarity Dim rst As ADODB.RecordSet Set rst = New ADODB.RecordSet 'More code here If intValue > 1000 Then

'Raise an Error------------------------------------------------------> Set rst = Nothing On Error Goto 0 Call Err.Raise(Number:=lngcInvalidValue, \_

Description:=strcInvalidValue) End if Set rst = Nothing

End Sub

In the example just shown, if the rst object variable is not set to Nothing before performing the Err.Raise, the Class\_Terminate of the rst object likely will change the properties of the Err object so that it no longer reflects the number given in lngcInvalidValue. This Class\_Terminate code executes before the calling routine's error handler is invoked. This weird flow of execution has caused a number of very difficult to track down bugs.

Parameters to a Procedure

Every parameter to a procedure should be given an explicit data type, including variants. Every parameter should be passed by value using the ByVal keyword, with a few exceptions. These are:

* VBA doesn't allow certain data types to be passed by value, such as arrays, user-defined types, and objects.
* You specifically want to allow the changed value of the parameter to be passed back to the calling routine.
* The parameter to event procedure is specified as being by reference when VBA creates it.
* The arguments to a Declare statement must match the definition in the DLL.

Even in the cases where the argument should be passed by reference, you should explicitly prefix the parameter with ByRef, even though this is the default in VBA. This makes it explicit that you meant to pass that parameter by reference.

After VBA inserts an event procedure, the parameters to the event procedure should be changed to include ByVal and ByRef keywords, and change the parameter names to use the appropriate naming conventions. For example, VBA inserts the event procedure like this (with the line wrapped in this document):

Private Sub Form\_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

End Sub

This should be changed to read like this:

Private Sub Form\_MouseMove(ByRef intButton As Integer, \_ Byref intShift As Integer, ByRef sngX As Single, ByRef sngY As Single)

End Sub

By changing it to read like this, the naming conventions indicate the data type and the ByRef keywords indicate that VBA may see the changes to the parameters.

Parentheses

Always use parentheses where the reading of the line may be unclear. For example, suppose that a line is written:

If Not frmTest Is Nothing Then

It may not be clear that the Is operator has higher precedence than the Not operator in this line. Recode it to read:

If Not (frmTest Is Nothing) Then

This makes it clear what order the operators are evaluated. The general rule is that if there is any question what the operator precedence is, use parentheses to make it clear.

Procedure Scope

Always use the Private scope on a procedure unless you need to expose the procedure outside the current module. In a library, use the Friend scope when you need a larger scope. Use Public only when access to the procedure is required outside the library. For Example:

Private Sub Test()

Project Properties [VB6]

In Visual Basic, the Project Properties dialog should always be filled in. These values may not apply to VBA hosts other than Visual Basic. Most of these fields can be retrieved from the EXE, DLL, or OCX file by right clicking on it in the Windows Explorer, then selecting Properties, then clicking on the Version tab in the dialog that appears. The values can be retrieved from within the program by getting properties of the App global system object. The following fields should be always be filled in:

* Project Name: The name of the library or standard EXE name. The library name should always start with a short word or abbreviation indicating the company or organization that is developing the library. For example, the Maya Calendar engine library from Xoc Software, might be named XocEngine or XocMayaEngine. This term is used for disambiguation of libraries and shows in the Object Browser. See the Disambiguation section. This is the internal name of the library. This may have abbreviations in it.
* Project Description: This should be the same name as the Project Name, except with spaces between the words. Abbreviations and the company or organization name should be spelled out. For example, use Xoc Maya Engine. These words show up in the VBA References dialog.
* Major/Minor/Revision number: These should be filled in with appropriate values. The version number should never be set to a smaller value as installation programs depend on it to determine if they should overwrite an older version with a newer one.
* Auto Increment: In most cases this should be checked. This automatically increments the revision number by one every time the project is compiled to a file.
* Application Title: The application title should be the name of the product that you expect to show externally, on the Windows Start menu, the Windows task list, the Windows Task Bar, and should be copied to the Caption of the main form in the application when the program starts. For example: Xoc Maya Engine.
* Comments: If Visual SourceSafe is used to maintain the project, this should be filled in with $Header: $. If keyword expansion in files is used, then Visual SourceSafe will place the expansion into the comments section of the executable. This gives the source name of the project is, the SourceSafe version number of the VBP file, the date and time the project file was changed, and by whom. This helps roll back the project to a given release to test for bugs. See the section on Source Code Control to configure SourceSafe.
* Company Name: Should be filled with your company or organization name. For example: Xoc Software. This is used on splash screens and about dialogs. Therefore, if your company is XYZ Software, Inc., you probably want to use XYZ Software.
* File Description: This is the description of how this file fits into the entire package. For example: Xoc Maya Calendar calculation engine or Xoc Maya Calendar UI.
* LegalCopyright: Enter the copyright notice for the program. For example: Copyright © 1999 by Xoc Software. You may find it useful to type Alt+0169 on the keypad (not the main keyboard) to get the © symbol in the dialog.
* LegalTrademarks: Enter any trademarks or registered trademarks for the company or product. For example: Xoc™ is a trademark of Xoc Software. You may find it useful to type Alt+0153 on the keypad (not the main keyboard) to get the ™ symbol and Alt+0174 to get the ® symbol. Note that these symbols may or may not show correctly in the application depending on the font you choose to display them.
* Product Name: This is the name of the product, without the company name. Therefore, if the name of the product elsewhere is Xoc Maya Calendar, the name here should be just Maya Calendar. This value may be used in splash screens and about dialogs.

Raising Errors

When you raise a runtime error from a component, to be trapped in the calling code, the error number that you raise should have a unique error number. For this purpose, VBA defines a constant vbObjectError that guarantees that errors that you generate will not conflict with ones that VBA defines. However, all libraries use errors in the range larger than vbObjectError, so you should strive to be different from the other libraries with your numbers. There is no way to guarantee this; the chances can be reduced by starting your errors at a random number in the range 512 to 32767 larger than vbObjectError. No library that an organization produces should ever have conflicting error numbers with another library from the same organization. For example: XYZ Software might start numbering its errors at vbObjectError + 4096. The first library produced from XYZ software might generate errors in the range from vbObjectError + 4096 to vbObjectError + 4146, the second library from vbObjectError + 4147 to vbObjectError + 4196, etc.

Select/End Select Blocks

The Select/End Select block is indented differently from other blocks (see Indenting). The Case blocks within the Select/End Select are lined up with the Select/End Select keywords. Code within a Case block is indented one tab stop from the Case statement. For Example:

Select Case Err.Number

Case tliErrCantLoadLibrary

Err.Raise Number:=Err.Number, Description:=Err.Description, \_

Source:=Err.Source

Case 35602 'This key is already associated with an element of this

'collection

Set nodChild = tvw.Nodes.Item(cci.Guid)

nodChild.Image = "InstClass"

Resume NextItem

Case Else

Call UnexpectedError(Err.Number, Err.Description, Err.Source, \_

Err.HelpFile, Err.HelpContext)

End Select

In non-RVBA coding standards, it is more common to indent Case blocks one tab stop from the surrounding Select/End Select. However, this causes the actual executing code to be indented two tab stops from the surrounding Select/End Select. The readability of the code is just as good, if not better with this scheme, although it takes some getting use to the first Case block being indented to the same level as the Select line.

See also the note about Case Else blocks in the section on Assertions.

{Alternative: Indent the Case blocks one tab stop from the surrounding Select/End Select. Then indent the code in the case blocks one more tab stop.}

Source Code Control [VB6]

Code should be maintained using some sort of Source Code Control. Microsoft Visual SourceSafe is the most common product used for this. When using Visual SourceSafe, the Administrator should configure it to expand keywords in files in the SourceSafe Administrator Options dialog. The following files should be expanded: \*.bas,\*.cls,\*.ctl,\*.frm,\*.pag,\*.vbp. Entries such as $Header: $ can then be placed into the code and are expanded automatically. See the SourceSafe documentation on keyword expansion. Also, see the use of the Comments entry in the section on Project Properties in this document.

Type Conversion

VBA is considered a weakly typed language. You can construct expressions such as this one:

strValue = "Your order came to " & intQuantity \* curPrice

VBA will automatically convert the result of the expression into a string to make the expression work. However, it is better programming practice to make explicit what VBA is doing using the type conversion functions: CBool, CInt, CLng, CStr, etc., plus the Format$ function. For example:

strValue = "Your order came to " \_ & Format$(CCur(intQuantity) \* curPrice, "$#,###.00")

The RVBA naming conventions will help point out possible bugs. If you see a line that looks like this, you may have a potential bug:

intValue = lngInput

If the value in the variable lngInput is 90,000, this line will cause an Overflow runtime error. The fact that the types of the variables are different is a clear warning sign. If, however, you knew the value in lngInput could only be in the range 1 to 1000, it might be acceptable to do the assignment like this:

Debug.Assert lngValue >= 1 And Debug.Assert lngValue <= 1000 intValue = CLng(lngInput)

See also the section on Assertions.

Variable Declaration

Every variable should be explicitly declared. Using the Option Explicit keyword at the top of the module will have VBA enforce that. The VBA editor's Tools Options dialog has a setting that will make this be automatically inserted in all new modules.

Every variable should be given an explicit data type. This includes variants, which are the default. For example, a variant should be declared as:

Dim varValue As Variant

Rather than letting it be implicitly defined or declaring it as:

Dim varValue

Every variable should be declared on a line by itself. This precludes running into this bug:

Dim intValue, intTest As Integer

That declaration makes it obvious that the two variables were meant to be declared as integers, but the first variable is defined as a variant. If instead the declarations were made on a line by themselves, the problem goes away. For example:

Dim intValue As Integer Dim intTest As Integer

In addition, by declaring each variable on a line by itself, you can use the Ctrl+Y keyboard shortcut to cut the declaration to the clipboard regardless of where the caret is on the line, then paste it somewhere else. If there are multiple declarations on a line then editing is not as easy.

Variable Initialization

Some languages allow declaring a variable and giving it an initial value at the same time. Visual Basic doesn't allow that. A variable has a default value at the time that it is declared based on its data type. However, a syntax variation can be used inside of a procedure to give the feel of initializing it and assigning the default value.

Visual Basic allows you to place multiple logical lines of code on the same physical line if you separate them with colons. It also allows you to mix the declaration of variables with executable lines of code. Therefore, inside a procedure you can initialize a variable and give it a default value in one physical line like this:

Public Sub Test()

'Error handling omitted for clarity

Dim intValue As Integer: intValue = 7

Dim strTest As String: strTest = "Default Value"

'other code End Sub

Variables Scope and Lifetime

Variables should always be declared with the smallest level of scope and the shortest lifetime possible. Thus, you should declare variables with Dim inside of procedures by preference. If you need a longer lifetime, then use Static. If you need a wider scope, use Private. Only use Public as a last resort. Public variables declared in standard modules are global and can be changed by any piece of code throughout the entire project. This makes debugging changes to their value very difficult. Global variables of this sort should only be used in the context where they are set during initialization of the program, then remain static for the rest of the time the program executing.

Global variables should never be changed in one part of the program to be retrieved in another part of the program. In such cases, you should use parameters of procedures or properties of forms or objects to pass the information. If there are more than 20 global variables in the program, it is a warning sign that the program design is wrong.

Having Public constants are allowed and encouraged. See the section on constants.

Version Compatibility [VB6]

After building an ActiveX control or ActiveX DLL for the first time in Visual Basic, the project compatibility should be set to Binary Compatibility in the Project Properties dialog. Each time the component is "released," a copy of the component should be made to the same directory, but with the filename extension set to CMP. The binary compatible file entry should point to this file. That means that you can modify the interface to the file within a release as long as you are still backwards compatible with the last release. The CMP file should be checked into the Source Code Control project, whereas the current copy of the component itself probably should not be checked in (see the section on Source Code Control).

*Greg Reddick is the President of Xoc Software, a software development company developing programs in Visual Basic, Microsoft Access, C/C++, and for the web. He leads training seminars in Visual Basic for Application Developers Training Company (AppDev). In a previous life, he worked for four years on the Access development team at Microsoft. Greg can be reached at mailto:grr@xoc.net or from the Xoc Software web site, http://www.xoc.net.*

APPENDIX E: MICROSOFT APPLICATION USER INTERFACE GUIDELINES

Intro

By using established guidelines we can ensure that we build our software products with appropriate and consistent appearances.

We will follow the standard Windows user interface guidelines as described in “*Windows User Interface Guide*”, Microsoft Press, 1999, and as described in the three documents that can be found at <http://sdg.jsi.com/standards/style-guides/user-interface/windows>. This document further specifies the guidelines set forth in those documents.

General Properties

Some properties are common to pages, controls, and elements. The rules set forth in this section should suffice in most cases.

Fonts

Face

When not defaulting to the system font, use Arial.

Size

Text

Use 8 point, normal.

Customer/Client

Use 12 point, normal.

Page Title

Use 14 point, bold.

Type

Use boldface type for titles. Use normal type elsewhere. Never use underlined or italic typefaces.

Color

All text should be black.

Background Color

The font background should always be transparent, thereby defaulting to the background color of the font’s parent form or control.

Case

Use title case for all labels, names, and titles. Do not use either all UPPERCASE or all lowercase.

Color

Palette

Deliver has selected a palette of five colors: red, orange, gold, green, and blue.

|  |  |  |
| --- | --- | --- |
| Red | HSL: 234, 195, 85  PMS: 194  RGB (decimal): 164, 17, 40  RGB (hexadecimal): #A41128  Windows: 2625956 |  |
| Orange | HSL: 20, 40, 120  PMS: 152  RGB (decimal): 255,125,0  RGB (hexadecimal): #FF7D00  Windows: 32255 | deliver-orange |
| Gold | HSL: 25, 240, 129  PMS: 137  RGB (decimal): 255, 168, 20  RGB (hexadecimal): #FFA814  Windows: 1353983 |  |
| Green | HSL: 74, 143, 74  PMS: 363  RGB (decimal): 47, 126, 32  RGB (hexadecimal): #2F7E20  Windows: 2129455 |  |
| Blue | HSL: 138, 229, 64  PMS: 541  RGB (decimal): 3, 76, 133  RGB (hexadecimal): #034C85  Windows: 8735747 |  |

Text

All text should be black. For further info on text properties, see the section on “Fonts”.

Dimensions

Screen

Height

All screens will fit onto a medium resolution terminal, i.e., 600 pixels high. If at all possible, no data will be displayed below the viewable area of the screen. In other words, avoid vertical scrolling.

Width

All screens will fit onto a medium resolution terminal, i.e., 800 pixels wide. No data should ever be displayed to the right of the viewable area of the screen. In other words, horizontal scrolling is forbidden.

Regions

Header & Primary Navigation

The header and primary navigation area will span the entire width of the page, occupying no more than the top 75 pixels of the screens height.

Secondary Navigation

The secondary navigation area will run down the left side of the screen, below the header. Of this space, the secondary navigation bar will occupy no more than 15% of the screen’s width.

Content

The content area will fill the area from the bottom of the header to the top of the footer, and from the right edge of the secondary navigation area to the right edge of the screen.

Ordinarily, this region will be subdivided into other regions dictated by the application and the application’s purpose.

Footer

The footer will span the breadth of the page, occupying no more than the bottom 40 pixels of the screen’s height.

Controls

Controls should be of uniform dimensions.

Buttons

(Borrowed from GUI LNF Standards – DENR (Interact))

If the length of text for a series of command buttons in a dialog box is similar, make all the buttons in the dialog box the size of the largest button

If the text length for a series of command buttons in a dialog box varies, use two button sizes—one for shorter text and another for longer text. Do not use more than two different button sizes in a dialog box.

Text Boxes

(Borrowed from GUI LNF Standards – DENR (Interact))

Size text boxes to indicate the approximate length of the field. If you have text boxes of similar length, make them the same length unless you need to show the exact size of the field. If the length of the field can vary, use text boxes of the same length to minimize the number of unique margins on the screen.

Left align text boxes on the screen to minimize the number of different margins. If a particular text box has a long label, use a different margin for that text box. Limit the number of unique margins to two.

List Boxes

Show at least three, but no more than eight items in a list box at a time. If you have more items use a scroll bar to view the rest of the items.

Flow

Tabbing from control to control will go from top to bottom, then left to right.

Usability

Keyboard vs. Mouse

All applications will be fully usable without a mouse, i.e., all functionality will be readily accessible from the keyboard.

Section 508

All applications will be Section 508 compliant.

Performance

The system will always provide some visual feedback to the user as soon as possible. For desktop applications, this feedback must occur within one second. For web applications, the feedback must occur within five seconds.

Internationalization

Multi-Language support

All apps will provide Unicode (wide character) multi-language support. This may mean that dialogs & screens have to be somewhat auto-sizing.

Date & Time

All applications will use the system-defined date and time formats.

Applications

Aside from the general, overarching principles used to govern screen layout and design, the various types of applications have their own special constraints.

Desktop

Desktop applications are applications that can be run in standalone mode without the use of a browser. These are traditional Windows applications.

Page

Splash Screen

Each Deliver app will feature a splash screen of uniform layout (actual layout is to be determined by the communications group). This splash screen will be 300 pixels by 400 pixels. The splash screen will persist for 5 seconds. The splash screen will be available from Help 🡪 About <appname>.

Framework

Each application will feature a title bar, a menu bar, a tool bar, and a status bar.

(Microsoft Access-based applications will use the framework provided by Access.)

Title Bar

The title bar is the line at the very top of the application or dialog, just above the menu bar (if the menu bar is present). The title bar contains the minimize, maximize, and restore buttons.

Font

The font should be Arial 12, bold.

Color

The title bar should be either the system color, or the appropriate color from the Deliver palette. Which one?

Content

The title bar should read “Page Name – Application Name”.

Case

Titles should be in Proper Case, except in the case of logos.

Menu Bar

Standard menu bar names and positions will be used.

Tool Bars

Tool bars will use standard Windows icons.

Icon Size

There are two sizes, small and large. Small is 16x16 pixels, large is 20x20 pixels. Prefer the small icons.

Spacing

For each size, there should be 3 pixels between a toolbar button and its text label.

Status Bar

The status bar will be the Windows standard 1 line tall. It should be used to provide the user with information about the status of the application.

Sizing

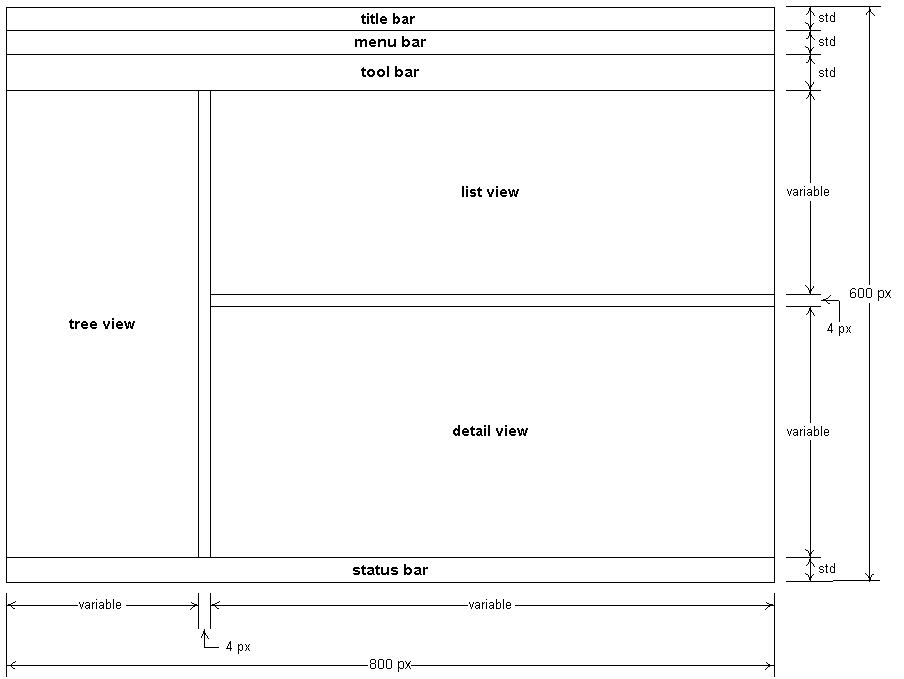
All screens will support minimizing, maximizing, and custom sizing.

Layout

General

Deliver uses “explorer” style applications. This provides for easiest navigation of the application’s screens.

The following drawing defines the recognized screen regions and their sizes.



Size

Target screen resolution: all apps should be presentable at 600 x 800.

Each pane in the window can be resized.

Windows will control the heights of standard Windows elements such as title bar, menu bar, tool bar, and status bar.

Font

Windows will control the fonts in the title bar, menu bar, tool bar, and status bar.

Color

Windows will control the colors in the title bar, menu bar, tool bar, and status bar.

Graphics

Splash Screen

Has the app logo, product version info, JSI logo, funding source (customer) logo, and application image.

About

Use the same info as the splash screen, add links to websites, email address.

Regions

The screen is divided into number of regions, each of which has a purpose.

Tree View

The Tree View is used to navigate the application.

Icons

We will use the following standard icons for the following purposes:

Form

Forms will be represented by a little picture of a form. (need sample)

Report

Reports will be represented by a piece of paper with writing on it. (need sample)

Question

Question marks are commonly used for this. (need sample)

Configuration

A picture of tools will represent configuration screens. (need sample)

Group/Section

This could be either plus sign for closed groups, and a minus sign for open groups, or opened and closed folders. Either set of icons is commonly used in Windows apps. (need sample)

Forms

Forms are essentially dialogs that do not pop up, but remain embedded in the application framework. Whereas dialogs will contain a title bar and minimize/maximize/close buttons, a forms do not.

Layout

For controls that do not contain their own labels, the label should be placed to the left or above the related control. This makes it easier for users to associate the label with the corresponding control.

The logo should go in the upper right corner.

All forms will support resizing, minimizing, and maximizing.

Logos

All Deliver sw has a logo area in top right area of each screen. This logo area will be used to brand the product as a Deliver app. This area will likely contain only the application’s logo. The JSI and customer logos would appear on the splash screen and in the “About” dialog.

Style (Special Effect)

Dialogs and forms should be flat.

Elements

Screen elements are the things that go in the regions. These are things like tool bars, labels, and titles.

Title Bar

This is the same as standard definition for a title bar, except that the dialog title bar should not include the “ – Application Name”.

Title

If a title is present it should be bold, 12pt, default font, proper case. No italics; no underline.

Menu Bar

Use standard menu bar.

Tool Bar

Use standard tool bar.

Labels

Use the default font.

Pop Up Dialogs

Modal

When to use

Modal dialogs are used whenever the application absolutely cannot continue without user input. In most cases, though, this can be handled with an “Apply” button.

Buttons

As a minimum, modal dialogs will have “OK”, “Cancel”, and “Help” buttons.

Modeless

When to use

Modeless dialogs are to be used unless the criteria for using a modal dialog are met.

Buttons

As a minimum, modeless dialogs will have “OK”, “Cancel”, and “Help” buttons. “Apply” is almost meaningless – the user must hit either “OK” or “Cancel” to close the dialog, and the functionality of “OK” is a superset of that of “Apply”.

Tabbed

When to use

Use tabbed dialogs when the dialog uses more controls than can be fit on a single screen.

Layout

Each tab should contain controls that relate to a particular dialog subtopic. If a control does not pertain to the topic of the overall dialog, then it should not be included on any of the tabs.

Buttons

Use consistent tab width, allowing the longest tab label dictate the tab width.

Each page of the tabbed dialog will feature, as a minimum, “OK”, “Cancel”, and “Apply” buttons. The buttons will follow the properties set forth in the General Properties section of this document.

Elements

Title Bar

Same as standard definition for a title bar, except that the dialog title bar should not include the “ – Application Name”.

Title

If a title is present it should be bold, 12pt, default font, proper case. No italics; no underline.

Menu Bar

Do not use menu bars in dialogs except in exceptional circumstances.

Tool Bar

Do not use tool bars on dialogs except in exceptional circumstances.

Labels

Use the default font.

Style (Special Effect)

Dialogs should be flat.

Reports

Size

DP-Body textFormat

Paper Size

DP-Body textMargins

DP-Body textLayout

DP-Body text

Controls

DP-Body textGeneral

Dialog Base Units

DP-Body textSize

DP-Body textExamples:

DP-Body textSpacing

DP-Body textControls

DP-Body textThe absolute smallest space between controls is 2 DLUs.

Dialogs

There should be a 7 DLUs between the edge of the dialog box and the text or frame.

Paragraphs

There should be 7 DLUs between paragraphs of text.

Text Labels

There should be 3 DLUs between text labels and their controls.

Group Boxes

The first control in a group box should be 11 DLUs down from the top of said group box.

Controls in a group box should be aligned vertically to the group box title.

The last control in a group box should be 7 DLUs above the bottom of the group box.

Buttons

If a text label is beside a button, it should be 3 DLUs down from the top of the button.

A check box, list box, or option button beside a button should be 2 DLUs down from the top of the button.

Grouping

The following rules apply to grouping:

* Group related components
* Group box controls
* Use separator lines on menus

Much more comprehensive discussions of grouping are available in the resources cited in the introduction to this document.

Color

Color can be used, but is not recommended. Allow the user to change color schemes.

Other considerations

Main command buttons in a secondary window should be stacked in the upper right corner or in a row along the bottom. If there is a default button, it should always be the first one in the set. OK and Cancel buttons should be placed next to each other.

Alignment

In group boxes, controls should be left-aligned with the text label of the group.

Command buttons in the group should be right-aligned.

In toolbar arrangements, buttons and other controls are typically left-aligned or top-aligned

Text box

Layout

Use a consistent width between boxes. Flow should be from top to bottom, left to right.

Unlocked

Font

Use the standard default.

Color

Text

Use the standard text color.

Background

White is the standard Windows color for the background of unlocked text boxes. We will let the system control this color.

Size

Text boxes should be of uniform length, unless there is a requirement to show the user how large the data field is.

Locked

Font

Use Deliver default.

Color

Text

Use the standard text color.

Background

The Windows standard color for locked text boxes is light gray, but we will let the system control this color.

Size

Text boxes should be of uniform length, unless there is a requirement to show the user how large the data field is.

List box

Font

Use default.

Color

Use default.

Size

Display from 3 to 8 rows. Allow vertical scrolling if there are more than 8 items on the list.

Avoid horizontal scrolling.

Combo box

Font

Use default.

Color

Use default.

Size

Display from 3 to 8 rows. Allow vertical scrolling if there are more than 8 items on the list.

Avoid horizontal scrolling.

Radio buttons

When to use

Use option buttons when users should pick one mutually exclusive choice from a list of options, for example, choosing a pay period in a personnel application.

Colors

Let the system decide.

Size

Use default.

Arrangement

Lay these out vertically, using an outline to group them.

Check boxes

When to use

Use check boxes when users can choose one or more options, but these choices are not mutually exclusive.

Use check boxes for toggling a single value on or off. It is okay to have just one check box.

Colors

Let the system decide.

Size

Use default.

Arrangement

Lay these out vertically, using an outline to group them.

Command Buttons

Size

See the description of Buttons in General/Dimensions/Buttons.

Behavior

Ok

Saves settings and closes screen.

Cancel

Discards changes and closes screen.

Apply

Saves changes and keeps user on current screen.

Clear

Discards changes and keeps user on current screen.

Next

Takes user to the next logical screen. Does not implicitly save changes.

Previous

Takes user to the next logical screen. Does not implicitly save changes.

Help

Displays help text.

For more information, please visit deliver.jsi.com.

USAID | DELIVER PROJECT

John Snow, Inc.

1616 Fort Myer Drive, 11th Floor

Arlington, VA 22209 USA

Phone: 703-528-7474

Fax: 703-528-7480

Email: ask[deliver@jsi.com](mailto:deliver@jsi.com)

Internet: deliver.jsi.com