

# FileMaker® Pro 15

## Functions Reference



© 2007–2016 FileMaker, Inc. All Rights Reserved.

FileMaker, Inc.  
5201 Patrick Henry Drive  
Santa Clara, California 95054

FileMaker and FileMaker Go are trademarks of FileMaker, Inc. registered in the U.S. and other countries. The file folder logo and FileMaker WebDirect are trademarks of FileMaker, Inc. All other trademarks are the property of their respective owners. FileMaker documentation is copyrighted. You are not authorized to make additional copies or distribute this documentation without written permission from FileMaker. You may use this documentation solely with a valid licensed copy of FileMaker software.

All persons, companies, email addresses, and URLs listed in the examples are purely fictitious and any resemblance to existing persons, companies, email addresses, or URLs is purely coincidental. Credits are listed in the Acknowledgments documents provided with this software. Mention of third-party products and URLs is for informational purposes only and constitutes neither an endorsement nor a recommendation. FileMaker, Inc. assumes no responsibility with regard to the performance of these products.

For more information, visit our website at <http://www.filemaker.com>.

Edition: 01

# Contents

|  |    |
|--|----|
| Script steps reference (alphabetical list) . . . . . | 10 |
| About functions . . . . .                            | 16 |
| Working with formulas and functions . . . . .        | 16 |
| Using this functions reference . . . . .             | 16 |
| Aggregate functions . . . . .                        | 17 |
| Average . . . . .                                    | 18 |
| Count . . . . .                                      | 19 |
| List . . . . .                                       | 20 |
| Max . . . . .  | 22 |
| Min . . . . .  | 24 |
| StDev . . . . .                                      | 26 |
| StDevP . . . . .                                     | 28 |
| Sum . . . . .  | 30 |
| Variance . . . . .                                   | 32 |
| VarianceP . . . . .                                  | 34 |
| Container functions . . . . .                        | 36 |
| Base64Decode . . . . .                               | 37 |
| Base64Encode . . . . .                               | 38 |
| GetContainerAttribute . . . . .                      | 39 |
| GetHeight . . . . .                                  | 44 |
| GetThumbnail . . . . .                               | 45 |
| GetWidth . . . . .                                   | 46 |
| VerifyContainer . . . . .                            | 47 |
| Date functions . . . . .                             | 48 |
| Date . . . . .                                       | 49 |
| Day . . . . .  | 50 |
| DayName . . . . .                                    | 51 |
| DayNameJ . . . . .                                   | 52 |
| DayOfWeek . . . . .                                  | 53 |
| DayOfYear . . . . .                                  | 54 |
| Month . . . . .                                      | 55 |
| MonthName . . . . .                                  | 56 |
| MonthNameJ . . . . .                                 | 57 |
| WeekOfYear . . . . .                                 | 58 |
| WeekOfYearFiscal . . . . .                           | 59 |
| Year . . . . .                                       | 60 |
| YearName . . . . .                                   | 61 |
| Design functions . . . . .                           | 62 |
| DatabaseNames . . . . .                              | 64 |
| FieldBounds . . . . .                                | 65 |
| FieldComment . . . . .                               | 66 |
| FieldIDs . . . . .                                   | 67 |
| FieldNames . . . . .                                 | 68 |
| FieldRepetitions . . . . .                           | 69 |

|                                  |           |
|----------------------------------|-----------|
| FieldStyle                       | 70        |
| FieldType                        | 71        |
| GetNextSerialValue               | 72        |
| LayoutIDs                        | 73        |
| LayoutNames                      | 74        |
| LayoutObjectNames                | 75        |
| RelationInfo                     | 76        |
| ScriptIDs                        | 78        |
| ScriptNames                      | 79        |
| TableIDs                         | 80        |
| TableNames                       | 81        |
| ValueListIDs                     | 82        |
| ValueListItems                   | 83        |
| ValueListNames                   | 84        |
| WindowNames                      | 85        |
| <b>External functions</b>        | <b>86</b> |
| External                         | 87        |
| <b>Financial functions</b>       | <b>88</b> |
| FV                               | 89        |
| NPV                              | 90        |
| PMT                              | 91        |
| PV                               | 92        |
| <b>Get functions</b>             | <b>93</b> |
| Get(AccountExtendedPrivileges)   | 98        |
| Get(AccountName)                 | 99        |
| Get(AccountPrivilegeSetName)     | 100       |
| Get(ActiveFieldContents)         | 101       |
| Get(ActiveFieldName)             | 102       |
| Get(ActiveFieldTableName)        | 103       |
| Get(ActiveLayoutObjectName)      | 104       |
| Get(ActiveModifierKeys)          | 105       |
| Get(ActivePortalRowNumber)       | 106       |
| Get(ActiveRepetitionNumber)      | 107       |
| Get(ActiveSelectionSize)         | 108       |
| Get(ActiveSelectionStart)        | 109       |
| Get(AllowAbortState)             | 110       |
| Get(AllowFormattingBarState)     | 111       |
| Get(ApplicationArchitecture)     | 112       |
| Get(ApplicationLanguage)         | 113       |
| Get(ApplicationVersion)          | 114       |
| Get(CalculationRepetitionNumber) | 115       |
| Get(ConnectionAttributes)        | 116       |
| Get(ConnectionState)             | 117       |
| Get(CurrentDate)                 | 118       |
| Get(CurrentExtendedPrivileges)   | 119       |
| Get(CurrentHostTimestamp)        | 120       |
| Get(CurrentPrivilegeSetName)     | 121       |
| Get(CurrentTime)                 | 122       |
| Get(CurrentTimestamp)            | 123       |
| Get(CurrentTimeUTCMilliseconds)  | 124       |

|                              |     |
|------------------------------|-----|
| Get(CustomMenuSetName)       | 125 |
| Get(DesktopPath)             | 126 |
| Get(Device)                  | 127 |
| Get(DocumentsPath)           | 128 |
| Get(DocumentsPathListing)    | 129 |
| Get(EncryptionState)         | 130 |
| Get(ErrorCaptureState)       | 131 |
| Get(FileMakerPath)           | 132 |
| Get(FileName)                | 133 |
| Get(FilePath)                | 134 |
| Get(FileSize)                | 135 |
| Get(FoundCount)              | 136 |
| Get(HighContrastColor)       | 137 |
| Get(HighContrastState)       | 138 |
| Get(HostApplicationVersion)  | 139 |
| Get(HostIPAddress)           | 140 |
| Get(HostName)                | 141 |
| Get(InstalledFMPlugins)      | 142 |
| Get(LastError)               | 143 |
| Get>LastExternalErrorDetail) | 144 |
| Get>LastMessageChoice)       | 145 |
| Get(LayoutAccess)            | 146 |
| Get(LayoutCount)             | 147 |
| Get(LayoutName)              | 148 |
| Get(LayoutNumber)            | 149 |
| Get(LayoutTableName)         | 150 |
| Get(LayoutViewState)         | 151 |
| Get(MenubarState)            | 152 |
| Get(ModifiedFields)          | 153 |
| Get(MultiUserState)          | 154 |
| Get(NetworkProtocol)         | 155 |
| Get(NetworkType)             | 156 |
| Get(PageNumber)              | 157 |
| Get(PersistentID)            | 158 |
| Get(PreferencesPath)         | 159 |
| Get(PrinterName)             | 160 |
| Get(QuickFindText)           | 161 |
| Get(RecordAccess)            | 162 |
| Get(RecordID)                | 163 |
| Get(RecordModificationCount) | 164 |
| Get(RecordNumber)            | 165 |
| Get(RecordOpenCount)         | 166 |
| Get(RecordOpenState)         | 167 |
| Get(RequestCount)            | 168 |
| Get(RequestOmitState)        | 169 |
| Get(ScreenDepth)             | 170 |
| Get(ScreenHeight)            | 171 |
| Get(ScreenScaleFactor)       | 172 |
| Get(ScreenWidth)             | 173 |
| Get(ScriptAnimationState)    | 174 |
| Get(ScriptName)              | 175 |
| Get(ScriptParameter)         | 176 |
| Get(ScriptResult)            | 177 |
| Get(SortState)               | 178 |

|                            |            |
|----------------------------|------------|
| Get(StatusAreaState)       | 179        |
| Get(SystemDrive)           | 180        |
| Get(SystemIPAddress)       | 181        |
| Get(SystemLanguage)        | 182        |
| Get(SystemNICAddress)      | 183        |
| Get(SystemPlatform)        | 184        |
| Get(SystemVersion)         | 185        |
| Get(TemporaryPath)         | 186        |
| Get(TextRulerVisible)      | 187        |
| Get(TotalRecordCount)      | 188        |
| Get(TouchKeyboardState)    | 189        |
| Get(TriggerCurrentPanel)   | 190        |
| Get(TriggerExternalEvent)  | 191        |
| Get(TriggerGestureInfo)    | 192        |
| Get(TriggerKeystroke)      | 194        |
| Get(TriggerModifierKeys)   | 195        |
| Get(TriggerTargetPanel)    | 197        |
| Get(UserCount)             | 198        |
| Get(Username)              | 199        |
| Get(UseSystemFormatsState) | 200        |
| Get(UUID)                  | 201        |
| Get(WindowContentHeight)   | 202        |
| Get(WindowContentWidth)    | 204        |
| Get(WindowDesktopHeight)   | 205        |
| Get(WindowDesktopWidth)    | 206        |
| Get(WindowHeight)          | 207        |
| Get(WindowLeft)            | 208        |
| Get(WindowMode)            | 209        |
| Get(WindowName)            | 210        |
| Get(WindowOrientation)     | 211        |
| Get(WindowStyle)           | 212        |
| Get(WindowTop)             | 213        |
| Get(WindowVisible)         | 214        |
| Get(WindowWidth)           | 215        |
| Get(WindowZoomLevel)       | 216        |
| <b>Logical functions</b>   | <b>217</b> |
| Case                       | 218        |
| Choose                     | 219        |
| Evaluate                   | 220        |
| EvaluationError            | 222        |
| ExecuteSQL                 | 223        |
| GetAsBoolean               | 225        |
| GetAVPlayerAttribute       | 226        |
| GetField                   | 229        |
| GetFieldName               | 231        |
| GetLayoutObjectAttribute   | 232        |
| GetNthRecord               | 235        |
| If                         | 236        |
| IsEmpty                    | 237        |
| IsValid                    | 238        |
| IsValidExpression          | 239        |
| Let                        | 240        |

|                            |            |
|----------------------------|------------|
| Lookup                     | 243        |
| LookupNext                 | 244        |
| Self                       | 246        |
| <b>Mobile functions</b>    | <b>247</b> |
| Location                   | 248        |
| LocationValues             | 249        |
| RangeBeacons               | 250        |
| <b>Number functions</b>    | <b>252</b> |
| Abs                        | 253        |
| Ceiling                    | 254        |
| Combination                | 255        |
| Div                        | 256        |
| Exp                        | 257        |
| Factorial                  | 258        |
| Floor                      | 259        |
| Int                        | 260        |
| Lg                         | 261        |
| Ln                         | 262        |
| Log                        | 263        |
| Mod                        | 264        |
| Random                     | 265        |
| Round                      | 266        |
| SetPrecision               | 267        |
| Sign                       | 268        |
| Sqrt                       | 269        |
| Truncate                   | 270        |
| <b>Repeating functions</b> | <b>271</b> |
| Extend                     | 272        |
| GetRepetition              | 273        |
| Last                       | 274        |
| <b>Summary functions</b>   | <b>275</b> |
| GetSummary                 | 276        |
| <b>Text functions</b>      | <b>277</b> |
| Char                       | 279        |
| Code                       | 280        |
| Exact                      | 282        |
| Filter                     | 283        |
| FilterValues               | 284        |
| Furigana                   | 285        |
| GetAsCSS                   | 286        |
| GetAsDate                  | 287        |
| GetAsNumber                | 288        |
| GetAsSVG                   | 289        |
| GetAsText                  | 290        |
| GetAsTime                  | 291        |
| GetAsTimestamp             | 292        |
| GetAsURLEncoded            | 293        |

|                                  |            |
|----------------------------------|------------|
| GetValue                         | 294        |
| Hiragana                         | 295        |
| KanaHankaku                      | 296        |
| KanaZenkaku                      | 297        |
| KanjiNumeral                     | 298        |
| Katakana                         | 299        |
| Left                             | 300        |
| LeftValues                       | 301        |
| LeftWords                        | 302        |
| Length                           | 303        |
| Lower                            | 304        |
| Middle                           | 305        |
| MiddleValues                     | 306        |
| MiddleWords                      | 307        |
| NumToJText                       | 308        |
| PatternCount                     | 309        |
| Position                         | 310        |
| Proper                           | 311        |
| Quote                            | 312        |
| Replace                          | 313        |
| Right                            | 314        |
| RightValues                      | 315        |
| RightWords                       | 316        |
| RomanHankaku                     | 317        |
| RomanZenkaku                     | 318        |
| SerialIncrement                  | 319        |
| Substitute                       | 320        |
| Trim                             | 321        |
| TrimAll                          | 322        |
| Upper                            | 324        |
| ValueCount                       | 325        |
| WordCount                        | 326        |
| <b>Text formatting functions</b> | <b>327</b> |
| RGB                              | 328        |
| TextColor                        | 329        |
| TextColorRemove                  | 330        |
| TextFont                         | 331        |
| TextFontRemove                   | 332        |
| TextFormatRemove                 | 333        |
| TextSize                         | 334        |
| TextSizeRemove                   | 335        |
| TextStyleAdd                     | 336        |
| TextStyleRemove                  | 338        |
| <b>Time functions</b>            | <b>340</b> |
| Hour                             | 341        |
| Minute                           | 342        |
| Seconds                          | 343        |
| Time                             | 344        |
| <b>Timestamp functions</b>       | <b>345</b> |
| Timestamp                        | 346        |



|                               |     |
|-------------------------------|-----|
| Trigonometric functions ..... | 347 |
| Acos .....                    | 348 |
| Asin .....                    | 349 |
| Atan .....                    | 350 |
| Cos .....                     | 351 |
| Degrees .....                 | 352 |
| Pi .....                      | 353 |
| Radians .....                 | 354 |
| Sin .....                     | 355 |
| Tan .....                     | 356 |
| <br>                          |     |
| Glossary .....                | 357 |

## Script steps reference (alphabetical list)

|                                |     |
|--------------------------------|-----|
| Abs                            | 253 |
| Acos                           | 348 |
| Asin                           | 349 |
| Atan                           | 350 |
| Average                        | 18  |
| Base64Decode                   | 37  |
| Base64Encode                   | 38  |
| Case                           | 218 |
| Ceiling                        | 254 |
| Char                           | 279 |
| Choose                         | 219 |
| Code                           | 280 |
| Combination                    | 255 |
| Cos                            | 351 |
| Count                          | 19  |
| DatabaseNames                  | 64  |
| Date                           | 49  |
| Day                            | 50  |
| DayName                        | 51  |
| DayNameJ                       | 52  |
| DayOfWeek                      | 53  |
| DayOfYear                      | 54  |
| Degrees                        | 352 |
| Div                            | 256 |
| Evaluate                       | 220 |
| EvaluationError                | 222 |
| Exact                          | 282 |
| ExecuteSQL                     | 223 |
| Exp                            | 257 |
| Extend                         | 272 |
| External                       | 87  |
| Factorial                      | 258 |
| FieldBounds                    | 65  |
| FieldComment                   | 66  |
| FieldIDs                       | 67  |
| FieldNames                     | 68  |
| FieldRepetitions               | 69  |
| FieldStyle                     | 70  |
| FieldType                      | 71  |
| Filter                         | 283 |
| FilterValues                   | 284 |
| Floor                          | 259 |
| Furigana                       | 285 |
| FV                             | 89  |
| Get(AccountExtendedPrivileges) | 98  |
| Get(AccountName)               | 99  |
| Get(AccountPrivilegeSetName)   | 100 |
| Get(ActiveFieldContents)       | 101 |
| Get(ActiveFieldName)           | 102 |
| Get(ActiveFieldTableName)      | 103 |
| Get(ActiveLayoutObjectName)    | 104 |

|                                  |     |
|----------------------------------|-----|
| Get(ActiveModifierKeys)          | 105 |
| Get(ActivePortalRowNumber)       | 106 |
| Get(ActiveRepetitionNumber)      | 107 |
| Get(ActiveSelectionSize)         | 108 |
| Get(ActiveSelectionStart)        | 109 |
| Get(AllowAbortState)             | 110 |
| Get(AllowFormattingBarState)     | 111 |
| Get(ApplicationArchitecture)     | 112 |
| Get(ApplicationLanguage)         | 113 |
| Get(ApplicationVersion)          | 114 |
| Get(CalculationRepetitionNumber) | 115 |
| Get(ConnectionAttributes)        | 116 |
| Get(ConnectionState)             | 117 |
| Get(CurrentDate)                 | 118 |
| Get(CurrentExtendedPrivileges)   | 119 |
| Get(CurrentHostTimestamp)        | 120 |
| Get(CurrentPrivilegeSetName)     | 121 |
| Get(CurrentTime)                 | 122 |
| Get(CurrentTimestamp)            | 123 |
| Get(CurrentTimeUTCMilliseconds)  | 124 |
| Get(CustomMenuSetName)           | 125 |
| Get(DesktopPath)                 | 126 |
| Get(Device)                      | 127 |
| Get(DocumentsPath)               | 128 |
| Get(DocumentsPathListing)        | 129 |
| Get(EncryptionState)             | 130 |
| Get(ErrorCaptureState)           | 131 |
| Get(FileMakerPath)               | 132 |
| Get(FileName)                    | 133 |
| Get(FilePath)                    | 134 |
| Get(FileSize)                    | 135 |
| Get(FoundCount)                  | 136 |
| Get(HighContrastColor)           | 137 |
| Get(HighContrastState)           | 138 |
| Get(HostApplicationVersion)      | 139 |
| Get(HostIPAddress)               | 140 |
| Get(HostName)                    | 141 |
| Get(InstalledFMPlugins)          | 142 |
| Get(LastError)                   | 143 |
| Get>LastExternalErrorDetail)     | 144 |
| Get>LastMessageChoice)           | 145 |
| Get(LayoutAccess)                | 146 |
| Get(LayoutCount)                 | 147 |
| Get(LayoutName)                  | 148 |
| Get(LayoutNumber)                | 149 |
| Get(LayoutTableName)             | 150 |
| Get(LayoutViewState)             | 151 |
| Get(MenubarState)                | 152 |
| Get(ModifiedFields)              | 153 |
| Get(MultiUserState)              | 154 |
| Get(NetworkProtocol)             | 155 |
| Get(NetworkType)                 | 156 |
| Get(PageNumber)                  | 157 |
| Get(PersistentID)                | 158 |

|                              |     |
|------------------------------|-----|
| Get(PreferencesPath)         | 159 |
| Get(PrinterName)             | 160 |
| Get(QuickFindText)           | 161 |
| Get(RecordAccess)            | 162 |
| Get(RecordID)                | 163 |
| Get(RecordModificationCount) | 164 |
| Get(RecordNumber)            | 165 |
| Get(RecordOpenCount)         | 166 |
| Get(RecordOpenState)         | 167 |
| Get(RequestCount)            | 168 |
| Get(RequestOmitState)        | 169 |
| Get(ScreenDepth)             | 170 |
| Get(ScreenHeight)            | 171 |
| Get(ScreenScaleFactor)       | 172 |
| Get(ScreenWidth)             | 173 |
| Get(ScriptAnimationState)    | 174 |
| Get(ScriptName)              | 175 |
| Get(ScriptParameter)         | 176 |
| Get(ScriptResult)            | 177 |
| Get(SortState)               | 178 |
| Get(StatusAreaState)         | 179 |
| Get(SystemDrive)             | 180 |
| Get(SystemIPAddress)         | 181 |
| Get(SystemLanguage)          | 182 |
| Get(SystemNICAddress)        | 183 |
| Get(SystemPlatform)          | 184 |
| Get(SystemVersion)           | 185 |
| Get(TemporaryPath)           | 186 |
| Get(TextRulerVisible)        | 187 |
| Get(TotalRecordCount)        | 188 |
| Get(TouchKeyboardState)      | 189 |
| Get(TriggerCurrentPanel)     | 190 |
| Get(TriggerExternalEvent)    | 191 |
| Get(TriggerGestureInfo)      | 192 |
| Get(TriggerKeystroke)        | 194 |
| Get(TriggerModifierKeys)     | 195 |
| Get(TriggerTargetPanel)      | 197 |
| Get(UserCount)               | 198 |
| Get(UserName)                | 199 |
| Get(UseSystemFormatsState)   | 200 |
| Get(UUID)                    | 201 |
| Get(WindowContentHeight)     | 202 |
| Get(WindowContentWidth)      | 204 |
| Get(WindowDesktopHeight)     | 205 |
| Get(WindowDesktopWidth)      | 206 |
| Get(WindowHeight)            | 207 |
| Get(WindowLeft)              | 208 |
| Get(WindowMode)              | 209 |
| Get(WindowName)              | 210 |
| Get(WindowOrientation)       | 211 |
| Get(WindowStyle)             | 212 |
| Get(WindowTop)               | 213 |
| Get(WindowVisible)           | 214 |
| Get(WindowWidth)             | 215 |

|                          |     |
|--------------------------|-----|
| Get(WindowZoomLevel)     | 216 |
| GetAsBoolean             | 225 |
| GetAsCSS                 | 286 |
| GetAsDate                | 287 |
| GetAsNumber              | 288 |
| GetAsSVG                 | 289 |
| GetAsText                | 290 |
| GetAsTime                | 291 |
| GetAsTimestamp           | 292 |
| GetAsURLEncoded          | 293 |
| GetAVPlayerAttribute     | 226 |
| GetContainerAttribute    | 39  |
| GetField                 | 229 |
| GetFieldName             | 231 |
| GetHeight                | 44  |
| GetLayoutObjectAttribute | 232 |
| GetNextSerialValue       | 72  |
| GetNthRecord             | 235 |
| GetRepetition            | 273 |
| GetSummary               | 276 |
| GetThumbnail             | 45  |
| GetValue                 | 294 |
| GetWidth                 | 46  |
| Hiragana                 | 295 |
| Hour                     | 341 |
| If                       | 236 |
| Int                      | 260 |
| IsEmpty                  | 237 |
| IsValid                  | 238 |
| IsValidExpression        | 239 |
| KanaHankaku              | 296 |
| KanaZenkaku              | 297 |
| KanjiNumeral             | 298 |
| Katakana                 | 299 |
| Last                     | 274 |
| LayoutIDs                | 73  |
| LayoutNames              | 74  |
| LayoutObjectNames        | 75  |
| Left                     | 300 |
| LeftValues               | 301 |
| LeftWords                | 302 |
| Length                   | 303 |
| Let                      | 240 |
| Lg                       | 261 |
| List                     | 20  |
| Ln                       | 262 |
| Location                 | 248 |
| LocationValues           | 249 |
| Log                      | 263 |
| Lookup                   | 243 |
| LookupNext               | 244 |
| Lower                    | 304 |
| Max                      | 22  |
| Middle                   | 305 |

|                  |     |
|------------------|-----|
| MiddleValues     | 306 |
| MiddleWords      | 307 |
| Min              | 24  |
| Minute           | 342 |
| Mod              | 264 |
| Month            | 55  |
| MonthName        | 56  |
| MonthNameJ       | 57  |
| NPV              | 90  |
| NumToJText       | 308 |
| PatternCount     | 309 |
| Pi               | 353 |
| PMT              | 91  |
| Position         | 310 |
| Proper           | 311 |
| PV               | 92  |
| Quote            | 312 |
| Radians          | 354 |
| Random           | 265 |
| RangeBeacons     | 250 |
| RelationInfo     | 76  |
| Replace          | 313 |
| RGB              | 328 |
| Right            | 314 |
| RightValues      | 315 |
| RightWords       | 316 |
| RomanHankaku     | 317 |
| RomanZenkaku     | 318 |
| Round            | 266 |
| ScriptIDs        | 78  |
| ScriptNames      | 79  |
| Seconds          | 343 |
| Self             | 246 |
| SerialIncrement  | 319 |
| SetPrecision     | 267 |
| Sign             | 268 |
| Sin              | 355 |
| Sqrt             | 269 |
| StDev            | 26  |
| StDevP           | 28  |
| Substitute       | 320 |
| Sum              | 30  |
| TableIDs         | 80  |
| TableNames       | 81  |
| Tan              | 356 |
| TextColor        | 329 |
| TextColorRemove  | 330 |
| TextFont         | 331 |
| TextFontRemove   | 332 |
| TextFormatRemove | 333 |
| TextSize         | 334 |
| TextSizeRemove   | 335 |
| TextStyleAdd     | 336 |
| TextStyleRemove  | 338 |

|                        |     |
|------------------------|-----|
| Time .....             | 344 |
| Timestamp .....        | 346 |
| Trim .....             | 321 |
| TrimAll .....          | 322 |
| Truncate .....         | 270 |
| Upper .....            | 324 |
| ValueCount .....       | 325 |
| ValueListIDs .....     | 82  |
| ValueListItems .....   | 83  |
| ValueListNames .....   | 84  |
| Variance .....         | 32  |
| VarianceP .....        | 34  |
| VerifyContainer .....  | 47  |
| WeekOfYear .....       | 58  |
| WeekOfYearFiscal ..... | 59  |
| WindowNames .....      | 85  |
| WordCount .....        | 326 |
| Year .....             | 60  |
| YearName .....         | 61  |

# About functions

## Working with formulas and functions

A function is a predefined, named formula that performs a specific calculation and returns a single, specific value.

Most functions include three basic parts:

- the function
- a set of parentheses, if the function takes parameters
- the parameters required by the function

Each function returns a result of field type text, number, date, time, timestamp, or container.

A formula calculates a single value, based on constants (such as 1.07 or “hello”), operators (such as “+” or “>”), and field references (such as Subtotal or InvoiceTotal) you enter. For example, if sales tax in your area is 7% and you have a field named Subtotal, you could create a field named InvoiceTotal that gets the value of the formula `Subtotal * 1.07`.

If a formula is especially common or popular, FileMaker Pro gives it a name and defines exactly how you should use it. A named and predefined formula is called a function. For example, if you want to find the average of some test scores, you could write your own formula to add them all and divide by the total number of scores. A simpler approach would be to use the function named `Average` and follow the rules defined for its use.

## Using this functions reference

The content in this document was originally written for FileMaker Pro Help. It has been collected in this format to allow solution developers to read the information independent of the help system. Links to help topics may not work in this format.



## Aggregate functions

**Note** Aggregate [functions](#) perform statistical analysis on numbers (and dates or times for some functions) in:

- several [fields](#) in a [record](#).
- [related fields](#) whether displayed in a [portal](#) or not.
- [repeating fields](#).

For example, you can use the `Sum` function to add the values listed in a portal, as an alternative to creating a [report with grouped data](#) and subtotals.

The parameter values can include a numeric constant (for example, 10) or any valid expression. A constant parameter in a formula for a repeating field affects the result for every repetition.

When repeating field parameters (field1; field2;...) include a non-repeating field, that value is used in the result for only the first repetition unless you use the [Extend function](#).

Values in repetitions that exceed the number of repetitions in the calculated field are ignored. For example, a calculated field with three repetitions holds only three results, even when one field referenced in the calculation has five repetitions.

Click a function name for details.

| This function             | Returns  |
|---------------------------|--|
| <a href="#">Average</a>   | A value that is the average of all valid, non-blank values in a field.                         |
| <a href="#">Count</a>     | The number of valid, non-blank values in a field.  |
| <a href="#">List</a>      | A concatenated list of non-blank values for a field or fields.                                 |
| <a href="#">Max</a>       | The highest valid value in a field.  |
| <a href="#">Min</a>       | The smallest valid, non-blank value in a field.  |
| <a href="#">StDev</a>     | The standard deviation of the sample represented by a series of non-blank values in a field.   |
| <a href="#">StDevP</a>    | The standard deviation of a population represented by a series of non-blank values in a field. |
| <a href="#">Sum</a>       | The total of all valid, non-blank values in a field.   |
| <a href="#">Variance</a>  | The variance of a sample represented by a series of non-blank values in a field.               |
| <a href="#">VarianceP</a> | The variance of a population represented by a series of non-blank values in a field.           |

## Average

### Purpose

Returns a value that is the average of all valid, non-blank values in a field.

### Format

Average(field{;field...})

### Parameters

field - any [related field](#), [repeating field](#), or set of non-repeating [fields](#); or an [expression](#) that returns a field, repeating field, or set of non-repeating fields.

Parameters in braces { } are optional.

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Field can be any of the following:

- a repeating field (repeatingField).
- a field in matching [related records](#) specified by (table::field), whether or not these [records](#) appear in a [portal](#).
- several non-repeating fields in a record (field1;field2;field3...).
- corresponding repetitions of repeating fields in a record (repeatingField1;repeatingField2;repeatingField3), if the result is returned in a repeating field with at least the same number of repeats.
- several fields in the first matching record specified by (table::field1;table::field2;...). You can include fields from different [tables](#) (table 1::field A;table 2::field B...).

### Examples

A Student table has a portal showing scores for all exams a student has taken. The exam scores are in a table called Exams.

Average(Exams::Score) returns the student's average score for all exams she has taken.

In the following examples:

- Field1 contains two repetitions with values of 1 and 2.
- Field2 contains four repetitions with values of 5, 6, 7, and 8.
- Field3 contains 6.

Average(Field2) returns **6.5** when the calculation isn't a repeating field.

Average(Field1;Field2;Field3) returns **4, 4, 7, 8** when the calculation is a repeating field.

**Note** When a referenced field is a repeating field, the Average function returns the average of the values in the first repetition field, then the average of the values in the second repetition field, and so on. Therefore, **(1+5+6)/3=4;(2+6)/2=4;7/1=7;8/1=8**.

## Count

### Purpose

Returns the number of valid, non-blank values in a field.

### Format

Count (field{;field...})

### Parameters

field - any [related field](#), [repeating field](#), or set of non-repeating [fields](#); or an [expression](#) that returns a field, repeating field, or set of non-repeating fields.

Parameters in braces { } are optional.

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Field can be any of the following:

- a repeating field (repeatingField).
- a field in matching [related records](#) specified by (table::field), whether or not these records appear in a [portal](#).
- several non-repeating fields in a record (field1;field2;field3...).
- corresponding repetitions of repeating fields in a [record](#) (repeatingField1;repeatingField2;repeatingField3), if the result is returned in a repeating field with at least the same number of repeats.
- several fields in the first matching record specified by (table::field1;table::field2;...). You can include fields from different [tables](#) (table 1::field A;table 2::field B...).

### Examples

The Accounts [layout](#) has a portal showing installment payments made.

Count (Payments::Payment) returns the number of payments made on an account.

In the following examples:

- Field1 contains two repetitions with values of 1 and 2.
- Field2 contains four repetitions with values of 5, 6, 7, and 8.
- Field3 contains 6.

Count (Field2) returns **4** when the calculation isn't a repeating field.

Count (Field1;Field2;Field3) returns **3, 2, 1, 1** when the calculation is a repeating field.

**Note** When a referenced field is a repeating field, the Count function returns the total number of valid, non-blank values in the first repetition field, then the number of valid, non-blank values in the second repetition field, and so on.

## List

### Purpose

Returns a concatenated list of non-blank values for a field or fields.

### Format

```
List (field{ ; field... })
```

### Parameters

`field` - any [related field](#), [repeating field](#), or set of non-repeating [fields](#); an [expression](#) that returns a field, repeating field, or set of non-repeating fields, or a [variable](#).

Parameters in braces { } are optional.

### Data type returned

text

### Originated in

FileMaker Pro 8.5

### Description

Listed items are separated by carriage returns.

Use this function to return a list of values for:

- a single field (`table::field`), which returns a single result over all repetitions (if any) for this field and over all matching related records, whether or not these records appear in a portal.
- several fields and/or literal values (`table::field1, constant, table::field2...`), which returns a separate result for each repetition of the calculation across each corresponding repetition of the fields. If any fields are related, only the first related record is used.

### Examples

In the following examples:

- Field1 contains white.
- Field2 contains black.
- Field3 contains three repetitions with values of red, green, blue.
- Related::Field4 refers to three records that contain 100, 200, 300.
- \$f1 contains orange.

**Note** When referencing multiple repeating fields, `List ()` returns the list of values across the first repetition in the calculation's first repetition, then the list of values across the second repetition in the second repetition, and so on.

### Example 1

List (Field1; Field2) returns:

- white
- black

### Example 2

List (Field3) returns:

- red
- green
- blue

### Example 3

List (Field1; Field2; Field3) returns:

in calculation repetition 1:

- white
- black
- red

in calculation repetition 2:

- green

in calculation repetition 3:

- blue

### Example 4

List (Related::Field4) returns:

- 100
- 200
- 300

### Example 5

List (\$f1; Field2) returns:

- orange
- black

## Max

### Purpose

Returns the highest valid value in a field.

### Format

Max(field{;field...})

### Parameters

field - any [related field](#), [repeating field](#), or set of non-repeating [fields](#); or an [expression](#) that returns a field, repeating field, or set of non-repeating fields.

Parameters in braces { } are optional.

### Data type returned

text, number, date, time, timestamp

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Field can be any of the following:

- a repeating field (`repeatingField`).
- a field in matching [related records](#) specified by (`table::field`), whether or not these records appear in a [portal](#).
- several non-repeating fields in a [record](#) (`field1;field2;field3...`).
- corresponding repetitions of repeating fields in a record (`repeatingField1;repeatingField2;repeatingField3`), if the result is returned in a repeating field with at least the same number of repeats.
- several fields in the first matching record specified by (`table::field1;table::field2;...`). You can include fields from different [tables](#) (`table 1::field A;table 2::field B...`).

### Examples

The Accounts [layout](#) has a portal showing installment payments made.

Max(Payments::PaymentDate) returns the most recent date a payment was made on an account.

In the following examples:

- Field1 contains two repetitions with values of 1 and 2.
- Field2 contains four repetitions with values of 5, 6, 7, and 8.
- Field3 contains 6.

Max(Field2) returns **8** when the calculation isn't a repeating field.

Max(Field1;Field2;Field3) returns **6, 6, 7, 8** when the calculation is a repeating field.

## Notes

- When a referenced field is a repeating field, the `Max` function returns the maximum value in the first repetition field, then the maximum value in the second repetition field, and so on.
- Aggregate functions such as `Min` or `Max` use the data type of the first parameter to perform all comparisons. For example, if the first parameter's data type is text, all other parameters are converted to text and then compared.

## Min

### Purpose

Returns the smallest valid, non-blank value in a field.

### Format

Min(field{;field...})

### Parameters

field - any [related field](#), [repeating field](#), or set of non-repeating [fields](#); or an [expression](#) that returns a field, repeating field, or set of non-repeating fields.

Parameters in braces { } are optional.

### Data type returned

text, number, date, time, timestamp

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Field can be any of the following:

- a repeating field (repeatingField).
- a field in matching [related records](#) specified by (table::field), whether or not these records appear in a [portal](#).
- several non-repeating fields in a [record](#) (field1;field2;field3...).
- corresponding repetitions of repeating fields in a record (repeatingField1;repeatingField2;repeatingField3), if the result is returned in a repeating field with at least the same number of repeats.
- several fields in the first matching record specified by (table::field1;table::field2;...). You can include fields from different [tables](#) (table 1::field A;table 2::field B...).

### Examples

A Contracts table has a portal showing bids submitted for each contract.

Min(Bids::Price) returns the lowest bid submitted for a contract.

In the following examples:

- Field1 contains two repetitions with values of 1 and 2.
- Field2 contains four repetitions with values of 5, 6, 7, and 8.
- Field3 contains 6.

Min(Field2) returns **5** when the calculation isn't a repeating field.

Min(Field1;Field2;Field3) returns **1, 2, 7, 8** when the calculation is a repeating field.



## Notes

- When a referenced field is a repeating field, the `Min` function returns the minimum value in the first repetition field, then the minimum value in the second repetition field, and so on.
- Aggregate functions such as `Min` or `Max` use the data type of the first parameter to perform all comparisons. For example, if the first parameter's data type is text, all other parameters are converted to text and then compared.

## StDev

### Purpose

Returns the standard deviation of the sample represented by a series of non-blank values in a field.

### Format

StDev(field{;field...})

### Parameters

field - any [related field](#), [repeating field](#), or set of non-repeating [fields](#); or an [expression](#) that returns a field, repeating field, or set of non-repeating fields.

Parameters in braces { } are optional.

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Field can be any of the following:

- a repeating field (repeatingField).
- a field in matching [related records](#) specified by (table::field), whether or not these records appear in a [portal](#).
- several non-repeating fields in a [record](#) (field1;field2;field3).
- corresponding repetitions of repeating fields in a record (repeatingField1;repeatingField2;repeatingField3), if the result is returned in a repeating field with at least the same number of repeats.
- several fields in the first matching record specified by (table 1::field A, table 2::field B,...). You can name a different [table](#) for each field (table 1::field A;table 2::field B...).

$$\text{StDev} = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_n^2}{n-1} - \frac{(x_1 + x_2 + \dots + x_n)^2}{n(n-1)}}$$

## Examples

A portal displays the related values 5, 6, 7, and 8 in a field called Scores.

`StDev(table::Scores)` returns **1.29099444...**

In the following examples:

- Field1 contains two repetitions with values of 1 and 2.
- Field2 contains four repetitions with values of 5, 6, 7, and 8.
- Field3 contains four repetitions with values of 6, 0, 4, and 4.
- Field4 contains one repetition with a value of 3.

`StDev(Field4)` results in an error because standard deviation of a single number is not defined.

`StDev(Field1;Field2;Field3)` returns **2.64575131...**, **3.05505046...**, **2.12132034...**, **2.82842712...** for a repeating field.

**Note** When a referenced field is a repeating field, the `StDev` function returns the standard deviation in the first repetition fields, then the standard deviation in the second repetition fields, and so on.

## StDevP

### Purpose

Returns the standard deviation of a population represented by a series of non-blank values in a field.

### Format

StDevP(field{;field... })

### Parameters

field - any [related field](#), [repeating field](#), or set of non-repeating [fields](#); or an [expression](#) that returns a field, repeating field, or set of non-repeating fields.

Parameters in braces { } are optional.

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Field can be any of the following:

- a repeating field (repeatingField).
- a field in matching [related records](#) specified by (table::field), whether or not these records appear in a [portal](#).
- several non-repeating fields in a record (field1;field2;field3...).
- corresponding repetitions of repeating fields in a record (repeatingField1;repeatingField2;repeatingField3), if the result is returned in a repeating field with at least the same number of repeats.
- several fields in the first matching record specified by (table::field1;table::field2;...). You can include fields from different [tables](#) (table 1::field A;table 2::field B...).

$$\text{StDevP} = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_n^2}{n} - \left(\frac{x_1 + x_2 + \dots + x_n}{n}\right)^2}$$

## Examples

A portal displays the related values 5, 6, 7, and 8 in the field Scores. `StDevP(table::Scores)` returns **1.11803398...**

In the following examples:

- Field1 contains two repetitions with values of 1 and 2.
- Field2 contains four repetitions with values of 5, 6, 7, and 8.
- Field3 contains four repetitions with values of 6, 0, 4, and 4.
- Field4 contains one repetition with a value of 3.

`StDevP(Field4)` results in an error because the population standard deviation of a single number is not defined.

`StDevP(Field2)` returns **1.11803398...** for a non-repeating field.

`StDevP(Field1;Field2;Field3)` returns **2.16024689...**, **2.49443825...**, **1.5**, **2** for repeating fields.

**Note** When a referenced field is a repeating field, the `StDevP` function returns the standard deviation of a population in the first repetition fields, then the standard deviation of a population in the second repetition fields, and so on.

## Sum

### Purpose

Returns the total of all valid, non-blank values in a field.

### Format

```
Sum(field{;field...})
```

### Parameters

field - any [related field](#), [repeating field](#), or set of non-repeating [fields](#); or an [expression](#) that returns a field, repeating field, or set of non-repeating fields.

Parameters in braces { } are optional.

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Field can be any of the following:

- a repeating field ([repeatingField](#)).
- a field in matching [related records](#) specified by ([table::field](#)), whether or not these records appear in a [portal](#).
- several non-repeating fields in a [record](#) ([field1;field2;field3...](#)).
- corresponding repetitions of repeating fields in a record ([repeatingField1;repeatingField2;repeatingField3](#)), if the result is returned in a repeating field with at least the same number of repeats.
- several fields in the first matching record specified by ([table::field1;table::field2;...](#)). You can include fields from different [tables](#) ([table 1::field A;table 2::field B...](#)).

## Examples

An Invoice table has a portal showing line items.

`Sum(LineItems::ExtendedPrice)` totals the amounts for all items on the invoice.

A TimeBilling table has a portal showing time worked on a project. Hours is a time field.

`Sum(Hours::BillableHours)` returns the total number of billable hours on a project. Thus, if the portal shows **40** hours and **15:30** hours, the total billable hours are 55:30, or 55 1/2 hours.

In the following examples:

- Field1 contains two repetitions with values of 1 and 2.
- Field2 contains four repetitions with values of 5, 6, 7, and 8.
- Field3 contains 6.

If the calculation result isn't a repeating field:

- `Sum(Field2)` returns **26**.
- `Sum(Field1;Field2;Field3)` returns **12**.

If the calculation result is a repeating field:

- `Sum(Field2)` returns a repeating field with **26** in the first repetition.
- `Sum(Field1;Field2;Field3)` returns a repeating field with **12, 8, 7, 8**.

**Note** When a referenced field is a repeating field, the `Sum` function returns the sum of the first repetition field, then the sum of the second repetition field, and so on.

## Variance

### Purpose

Returns the variance of a sample represented by a series of non-blank values in a field.

### Format

Variance (field { ; field... })

### Parameters

field - any [related field](#), [repeating field](#), or set of non-repeating [fields](#); or an [expression](#) that returns a field, repeating field, or set of non-repeating fields.

Parameters in braces { } are optional.

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

The variance of a distribution is a measure of how spread out the distribution is. Field can be any of the following:

- a repeating field (repeatingField).
- a field in matching [related records](#) specified by (table::field), whether or not these records appear in a [portal](#).
- several non-repeating fields in a record (field1;field2;field3...).
- corresponding repetitions of repeating fields in a [record](#) (repeatingField1;repeatingField2;repeatingField3), if the result is returned in a repeating field with at least the same number of repeats.
- several fields in the first matching record specified by (table::field1;table::field2;...). You can include fields from different [tables](#) (table 1::field A;table 2::field B...).

$$\text{Variance} = \frac{x_1^2 + x_2^2 + \dots + x_n^2}{n - 1} - \frac{(x_1 + x_2 + \dots + x_n)^2}{n(n - 1)}$$



## Examples

A portal displays the related values 5, 6, 7, and 8 in Scores.

Variance(table::Scores) returns **1.66666666....**

In the following examples:

- Field1 contains two repetitions with values of 1 and 2.
- Field2 contains four repetitions with values of 5, 6, 7, and 8.
- Field3 contains four repetitions with values of 6, 0, 4, and 4.
- Field4 contains one repetition with a value of 3.

Variance(Field4) results in an error since the variance of a single value is not defined.

Variance(Field1;Field2;Field3) returns **7, 9.33333333...., 4.5, 8** if the calculation is a repeating field.

## Student example

Two classes of students take an exam. Class 1 has scores of 70, 71, 70, 74, 75, 73, 72 and Class 2 has scores of 55, 80, 75, 40, 65, 50, 95. The variance for each class is:

Class 1: **3.80952380...**

Class 2: **361.90476190...**

The variance for Class 1 is much lower than the variance for Class 2, because the scores for Class 2 are more spread out.

## VarianceP

### Purpose

Returns the variance of a population represented by a series of non-blank values in a field.

### Format

VarianceP(field{;field...})

### Parameters

field - any [related field](#), [repeating field](#), or set of non-repeating [fields](#); or an [expression](#) that returns a field, repeating field, or set of non-repeating fields.

Parameters in braces { } are optional.

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

The variance of a population distribution is a measure of how spread out the distribution is. Field can be any of the following:

- a repeating field (repeatingField).
- a field in matching [related records](#) specified by (table::field), whether or not these records appear in a portal.
- several non-repeating fields in a [record](#) (field1;field2;field3...).
- corresponding repetitions of repeating fields in a record (repeatingField1;repeatingField2;repeatingField3), if the result is returned in a repeating field with at least the same number of repeats.
- several fields in the first matching record specified by (table::field1;table::field2;...). You can include fields from different [tables](#) (table 1::field A;table 2::field B...).

$$\text{VarianceP} = \frac{x_1^2 + x_2^2 + \dots + x_n^2}{n} - \left( \frac{x_1 + x_2 + \dots + x_n}{n} \right)^2$$

## Examples

A [portal](#) displays the related values 5, 6, 7, and 8 in Scores.

VarianceP(table::Scores) returns **1.25**.

In the following examples:

- Field1 contains two repetitions with values of 1 and 2.
- Field2 contains four repetitions with values of 5, 6, 7, and 8.
- Field3 contains four repetitions with values of 6, 0, 4, and 4.
- Field4 contains one repetition with a value of 3.

VarianceP(Field4) results in an error since the variance of a single value is not defined.

VarianceP(Field1;Field2;Field3) returns **4.66666666...**, **6.22222222...**, **2.25**, **4** if the calculation is a repeating field.

## Student example

Two classes of students take an exam. Class 1 has scores of 70, 71, 70, 74, 75, 73, 72 and Class 2 has scores of 55, 80, 75, 40, 65, 50, 95. The population variance for each class is:

Class 1: **3.26530612...**

Class 2: **310.20408163...**

The population variance for Class 1 is much lower than the population variance for Class 2 because the scores for Class 1 are more tightly clustered.

## Container functions

Container [functions](#) calculate, manipulate, and report on data in container fields.

Click a function name for details.

| This function                         | Returns  |
|---------------------------------------|--|
| <a href="#">Base64Decode</a>          | Container content from text encoded in Base64 format.  |
| <a href="#">Base64Encode</a>          | The contents of the specified field as text in Base64 format.                                      |
| <a href="#">GetContainerAttribute</a> | The file metadata of the specified container field.  |
| <a href="#">GetHeight</a>             | The height, in pixels, of the content in a container field that holds images.                      |
| <a href="#">GetThumbnail</a>          | A thumbnail image of the content in a container field, according to a specified height and width.  |
| <a href="#">GetWidth</a>              | The width, in pixels, of the content in a container field that holds images.                       |
| <a href="#">VerifyContainer</a>       | 0 (false) if container data was changed or deleted outside FileMaker; otherwise, returns 1 (true). |

## Base64Decode

### Purpose

Returns container content from text encoded in Base64 format.

### Format

```
Base64Decode (text { ; fileNameWithExtension }
```

### Parameters

`text` - Base64 text to decode.

`fileNameWithExtension` - the filename and extension for the file created from the decoded Base64 text.

### Data type returned

container

### Originated in


FileMaker Pro 13.0

### Description

Base64 encoding does not retain the filename or extension of encoded content.

If a filename and extension are not specified in the `fileNameWithExtension` parameter, Base64Decode returns the container content with a generic filename and extension but does not change the content's data format.

### Examples

`Base64Decode (Products::Base64 ; "question.png")` returns  when `Products::Base64` is set to a string that begins with "iVBORw0KGgoAAAANSUhEUgAAAB8". The Base64 string in this example was shortened for readability.

## Base64Encode

### Purpose

Returns the contents of the specified field as text in Base64 format.

### Format

Base64Encode (sourceField)

### Parameters

sourceField - the name of a field.

### Data type returned

text

### Originated in

FileMaker Pro 13.0

### Description

Base64Encode adds a line break after every 76 characters.


Base64 encoding does not retain the filename or extension of encoded container field content.

### Example 1

Base64Encode (Products::Color) returns **QmxhY2s=** when Products::Color is set to "Black".

### Example 2

Base64Encode (Products::Container) returns a string that begins with

**iVBORw0KGgoAAAANSUhEUgAAAB8** when Products::Container is set to . The Base64 string in this example was shortened for readability.

## GetContainerAttribute

### Purpose

Returns the file metadata of the specified container field.

### Format

```
GetContainerAttribute (sourceField;attributeName)
```

### Parameters

`sourceField` - the name of a container field.

`attributeName` - the name of a supported attribute (see below).

### Data type returned

text, number, date, time, timestamp, container

### Originated in

FileMaker Pro 13.0

### Description

Some attributes may not return a result. For example, the values for the latitude and longitude of a photo may not be available, or some audio metadata like album art may not be available because the metadata is stored outside the audio file. Some individual attributes in the group attribute `all` may not be applicable in some circumstances.

**Tip** To minimize the frequency with which FileMaker Pro analyzes container data, use `GetContainerAttribute` in a field's auto-enter calculation and deselect **Do not replace existing value of field (if any)**. In this case, `GetContainerAttribute` will update only when the contents of the specified container field change. If `GetContainerAttribute` is used in a calculation field definition, the calculation may update each time the current record changes, reducing the performance of your solution.

### Attributes

| Attribute                 | Returns   | Data type returned |
|---------------------------|---|--------------------|
| <b>General</b>            |   |                    |
| <code>filename</code>     | The name of the file inserted into the container field.   | text               |
| <code>MD5</code>          | The result of applying the cryptographic hash function MD5 to a file inserted into the container field or a file referenced by a container field. | text               |
| <code>storageType</code>  | The method used to store the data in the container field:<br><b>Embedded, External (Secure), External (Open), File Reference, Text.</b>           | text               |
| <code>fileSize</code>     | The size (in bytes) of the file inserted into the container field.  | number             |
| <code>internalSize</code> | The amount (in bytes) of the space inside the database file that is occupied by the container field.  | number             |

| Attribute     | Returns  | Data type returned |
|---------------|--|--------------------|
| externalSize  | The amount (in bytes) of the space that is stored externally by the container field. This is either the size of the referenced file or the total size of all files in the container field (set up for open or secure storage).   | number             |
| externalFiles | A list of the external files associated with the container field (either files using open or secure storage or a file reference).  | text               |
| <b>Images</b> |  |                    |
| width         | A number representing the width of the image in pixels.  | number             |
| height        | A number representing the height of the image in pixels.   | number             |
| dpiWidth      | A number representing the horizontal DPI of the image.   | number             |
| dpiHeight     | A number representing the vertical DPI of the image.   | number             |
| transparency  | <b>1</b> if the image has an <a href="#">alpha channel</a> , otherwise returns <b>0</b> .  | number             |
| <b>Photos</b> |  |                    |
|               | <b>Note</b> Only JPEG and TIFF files return results.   |                    |
| orientation   | A number representing the orientation of the photo:<br><b>1 (Normal)</b><br><b>2 (Flipped horizontally)</b><br><b>3 (Rotated 180 degrees)</b><br><b>4 (Flipped vertically)</b><br><b>5 (Rotated 90 degrees counterclockwise and flipped vertically)</b><br><b>6 (Rotated 90 degrees counterclockwise)</b><br><b>7 (Rotated 90 degrees clockwise and flipped vertically)</b><br><b>8 (Rotated 90 degrees clockwise)</b><br><b>Note:</b> Photos that were inserted using earlier versions of FileMaker Pro are not automatically oriented; for such photos, <b>not applied</b> is appended to the result. For example, <b>3 (Rotated 180 degrees), not applied</b> . | text               |
| created       | The earliest available timestamp for the photo.  | timestamp          |
| modified      | The latest available timestamp for the photo. If the photo has never been modified, an empty string is returned.   | timestamp          |
| latitude      | The latitude of the location of the photo.   | text               |
| longitude     | The longitude of the location of the photo.  | text               |
| make          | The manufacturer of the camera used for the photo.   | text               |
| model         | The camera model used for the photo.   | text               |
| <b>Audio</b>  |  |                    |
|               | <b>Note</b> Only MP3 and M4A files return results.   |                    |
| title         | The title of the audio.  | text               |
| artist        | The name of the performer of the audio.  | text               |
| album         | The name of the album containing the audio.  | text               |



| Attribute         | Returns  | Data type returned |
|-------------------|--|--------------------|
| year              | The year the audio was released.   | text               |
| track             | The track number and count of the audio. For example, <b>3/12</b> , or <b>3</b> if the track count is not available. | text               |
| genre             | The genre of the audio.  | text               |
| composer          | The composer of the audio.   | text               |
| coverArt          | An image of the album cover.   | container          |
| duration          | The duration of the audio. For example, <b>0:03:16</b> .   | time               |
| bitRate           | The number of kilobits per second (kbps) used in the audio.  | number             |
| <b>Bar Codes</b>  |  |                    |
| barcodeText       | The contents of the bar code.  | text               |
| barcodeType       | The type of the bar code.  | text               |
| <b>Signatures</b> |  |                    |
| signed            | The timestamp when the signature was inserted.   | timestamp          |
| <b>Groups</b>     |  |                    |
| general           | Attributes listed in the General category above pertaining to the container field.                                   | text               |
| audio             | Attributes listed in the Audio category above pertaining to the container field.                                     | text               |
| image             | Attributes listed in the Images category above pertaining to the container field.                                    | text               |
| photo             | Attributes listed in the Photos category above pertaining to the container field.                                    | text               |
| barcode           | Attributes listed in the Bar Codes category above pertaining to the container field.                                 | text               |
| signature         | Attributes listed in the Signatures category above pertaining to the container field.                                | text               |
| all               | Attributes in all the categories listed above pertaining to the container field.                                     | text               |

## Notes

- The `internalSize` can be much smaller than the `fileSize` (for example, container fields set up for open or secure storage, file references, or compressed files) or much larger than the `fileSize` (for example, container fields created by plug-ins).
- Using the attribute `MD5` allows you to prevent the insertion of duplicated files into a container field regardless of the filename.
- Bar codes and signatures are not considered images.
- For the group attributes `general`, `audio`, `image`, `photo`, and `all`, attributes are displayed in the format `attributeName: attributeValue` with one attribute per line. Some attributes are displayed differently in order to fit the attribute on one line:
  - `externalFiles`. Displays only the number of external files.
  - `transparency`. Displays 1 (True) or 0 (False).
  - `coverArt`. Displays **png** or **jpg** depending on the type of image.
  - `bitRate`. Displays **kbps** after the number. If an audio uses a variable bit rate, **(VBR)** is appended to the result. For example: **Bit Rate: 247 kbps (VBR)**.
  - `year`. Date information may be returned in parentheses after the year. For example: **Year: 2014 (11/10/2014)**.

## Examples

Notice that the attributes in the following examples are enclosed in quotation marks.

`GetContainerAttribute(Image;"all")` returns:

### [General]

**Filename: IMG\_003.JPG**

**Storage Type: Embedded**

**MD5: C35A3F668A1FB3F370969399A1FF04FE**

**File Size: 1964978**

**Internal Size: 1965064**

**External Size: 0**

**External Files: 0**

### [Image]

**Width: 1936**

**Height: 2592**

**DPI Width: 72**

**DPI Height: 72**

**Transparency: 0 (False)**

**[Photo]**

**Orientation: 6 (Rotated 90 degrees counterclockwise)**

**Created: 1/14/2015 2:40:31 PM**

**Modified:**

**Latitude: 37.406167**

**Longitude: -121.983333**

**Make: Apple**

**Model: iPhone 5s**

`GetContainerAttribute(Product;"barcode")` returns:

**[Bar Code]**

**Bar Code Text: 875720001107**

**Bar Code Type: UPC-A**

`GetContainerAttribute(Package;"signature")` returns:

**[Signature]**

**Signed: 11/10/2014 11:41:22 AM**

## GetHeight

### Purpose

Returns the height, in pixels, of the content in a container field that holds images.

### Format

`GetHeight (field)`

### Parameters

`field` - any text, number, date, time, timestamp, or container field; or any text expression or numeric expression.

### Data type returned

number

### Originated in

FileMaker Pro 12.0

### Description

Returns the height in pixels of images in a container field that holds images. Otherwise, `GetHeight` returns 0.

### Examples

`GetHeight (product)` returns **768**.

## GetThumbnail

### Purpose

Returns a thumbnail image of the content in a container field, according to a specified height and width.

### Format

GetThumbnail (field;width;height)

### Parameters

*field* - any text, number, date, time, timestamp, or container field; or any text expression or numeric expression.

*width* - the width for the thumbnail.

*height* - the height for the thumbnail.

### Data type returned

container

### Originated in

FileMaker Pro 12.0

### Description

Returns an image that's stored in a container field according to specified values for width and height. The thumbnail image always maintains the proportions of the original image. Thumbnails cannot be made larger than the image's original size.

**Note** If the *field* parameter does not specify a field that contains image data, *field* must evaluate to the file path of an image. See [Creating file paths](#).

### Example 1

Exports a thumbnail with a maximum height and width of 50 points.

```
Set Field [Invoices::ExportContainer ; GetThumbnail ( Invoices::Container
; 50 ; 50 )
Export Field Contents [Invoices::ExportContainer]
```

### Example 2

Creates a thumbnail, exports it, and attaches it to an email.

```
Set Field [Invoices::ExportContainer; GetThumbnail ( Invoices::Container
; 50 ; 50 )
Set Variable [$ATTACHMENT; Value:Get ( TemporaryPath ) &
GetContainerAttribute ( Invoices::ExportContainer ; "filename" )]
Export Field Contents [Invoices::ExportContainer; "$ATTACHMENT"]
Send Mail [Send via E-mail Client; With dialog: Off; To: Customers::Email;
"$ATTACHMENT"]
```

## GetWidth

### Purpose

Returns the width, in pixels, of the content in a container field that holds images.

### Format

`GetWidth(field)`

### Parameters

*field* - any text, number, date, time, timestamp, or container field; or any text expression or numeric expression.

### Data type returned

number

### Originated in

FileMaker Pro 12.0

### Description

Returns the width in pixels of images in a container field that holds images. Otherwise, `GetWidth` returns 0.

### Examples

`GetWidth(Product)` returns **1024**.

## VerifyContainer

### Purpose

Returns 0 (false) if container data was changed or deleted outside FileMaker; otherwise, returns 1 (true).

### Format

VerifyContainer (field)

### Parameters

field - any text, number, date, time, timestamp, or [container field](#); or any text expression or numeric expression.

### Data type returned

text

### Originated in

FileMaker Pro 12.0

### Description

Returns a [Boolean](#) value representing the validity of data stored externally in a container field.

### Examples

VerifyContainer (Photo) returns:

- 0 (False) if files saved externally were modified or deleted.
- 1 (True) if no changes or deletions occurred.
- ? if the Photo field is not a container field.

## Date functions

Date [functions](#) calculate dates and manipulate date information.

---

**Important** To avoid errors when using dates, always use four-digit years. For more information about how FileMaker Pro handles two-digit dates, see [Conversion of dates with two-digit years](#).

---

**Note** [System formats](#) affect the way dates are displayed. See [Opening files with different system formats](#).

**Tip** You can use zero (0) and negative numbers as date function arguments. For example, the following formula returns **5/31/2014**:

```
Date (6;0;2014)
```

Click a function name for details.

| This function                    | Returns  |
|----------------------------------|--|
| <a href="#">Date</a>             | The calendar date for a month, day, and year.  |
| <a href="#">Day</a>              | A number from 1 to 31, for the day of the month on which a date occurs.              |
| <a href="#">DayName</a>          | Text that is the full name of the weekday for a date.                                |
| <a href="#">DayNameJ</a>         | Text in Japanese that is the full name of the weekday for a date.                    |
| <a href="#">DayOfWeek</a>        | A number representing the day of the week on which a date occurs.                    |
| <a href="#">DayOfYear</a>        | The number of days from the beginning of the year of a date.                         |
| <a href="#">Month</a>            | A number from 1 to 12, representing the month of the year in which a date occurs.    |
| <a href="#">MonthName</a>        | The full name of the month for a date.   |
| <a href="#">MonthNameJ</a>       | The name of the month of a date in Japanese.   |
| <a href="#">WeekOfYear</a>       | The number of weeks after January 1 of the year of a date.                           |
| <a href="#">WeekOfYearFiscal</a> | A number from 1 to 53, representing the week containing a date, from a starting day. |
| <a href="#">Year</a>             | A number representing the year in which a date occurs.                               |
| <a href="#">YearName</a>         | The Japanese year name of a date, in the specified format.                           |



## Date

### Purpose

Returns the calendar date for a month, day, and year.

### Format

Date (month;day;year)

### Parameters

month - the month of the year (a one-digit or two-digit number; see note).

day - the day of the month (a one-digit or two-digit number; see note).

year - the year (four digits between 0001 and 4000. For example, 2014 but not 14).

---

**Important** The order of the parameters in the Date function is always Month, Day, Year, no matter what operating system or FileMaker Pro date formats you are using.

---

### Data type returned

date

### Originated in

FileMaker Pro 6.0 or earlier

### Description

The format of the result depends on the date format that was in use when the database file was created. In the United States, dates are generally in the format MM/DD/YYYY. You can change the date format in your operating system.

You can change how the date is displayed by assigning a different date format to the field in [Layout mode](#). Changing the formatting in this way only affects the way the data is displayed, not how it is stored.

---

**Important** To avoid errors when using dates, always use four-digit years. For more information about how FileMaker Pro handles two-digit dates, see [Conversion of dates with two-digit years](#).

---

**Note** If you type a month greater than 12 or a day greater than the number of days in a month, FileMaker Pro adds the extra days or months to the result. The date function also allows zero and negative numbers as parameters. Decimal numbers are truncated to integers.

### Examples

Date (10;10;2014) returns **10/10/2014**.

Date (13;1;2014) returns **1/1/2015** (one month after December 1, 2014).

Date (6;0;2014) returns **5/31/2014** (one day before June 1, 2014).

Date (6;-2;2014) returns **5/29/2014** (three days before June 1, 2014).

Date (7;12;2014) -Date (7;2;2014) returns **10**.

"Bill Due by: " & Date (Month (DateSold) + 1;Day (DateSold);Year (DateSold)) returns **Bill Due by:** followed by a value that is one month later than DateSold.

# Day

## Purpose

Returns a number from 1 to 31, for the day of the month on which a date occurs.

## Format

Day (date)

## Parameters

date - any calendar date

## Data type returned

number

## Originated in

FileMaker Pro 6.0 or earlier

## Description

Use Day, for example, to identify the day of the month on which payments are due.

---

**Important** To avoid errors when using dates, always use four-digit years. For more information about how FileMaker Pro handles two-digit dates, see [Conversion of dates with two-digit years](#).

---

## Examples

Day ("5/15/2014") returns **15**. This example assumes that the system date format is MM/DD/YYYY.

Day (DateSold) returns the day of the month stored in DateSold.

If (Day (Get (CurrentDate)) = 15 and Month (Get (CurrentDate)) = 3; "Beware the Ides of March"; "") displays the text **Beware the Ides of March** only when the day of the month returned by Get (CurrentDate) is 15 and the month returned by Get (CurrentDate) is 3; otherwise it displays nothing.

## DayName

### Purpose

Returns text that is the full name of the weekday for a date.

### Format

DayName (date)

### Parameters

date - any calendar date

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

DayName (Date (10 ; 7 ; 2014) ) returns **Tuesday**.

DayName (ProjectDue) returns **Tuesday** when ProjectDue is 10/7/2014.

DayName ("10/7/2014" ) returns **Tuesday**.

"Return your selection by " & DayName (DueDate) displays the text **Return your selection by** followed by the name of the day stored in DueDate.

---

**Important** To avoid errors when using dates, always use four-digit years. For more information about how FileMaker Pro handles two-digit dates, see [Conversion of dates with two-digit years](#).

---

## DayNameJ

### Purpose

Returns text in Japanese that is the full name of the weekday for a date.

### Format

DayNameJ (date)

### Parameters

date - any calendar date

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

DayNameJ (Date (4 ; 4 ; 2014)) returns 金曜日 .

---

**Important** To avoid errors when using dates, always use four-digit years. For more information about how FileMaker Pro handles two-digit dates, see [Conversion of dates with two-digit years](#).

---

## DayOfWeek

### Purpose

Returns a number representing the day of the week on which a date occurs.

### Format

DayOfWeek (date)

### Parameters

date - any calendar date

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

The number 1 represents Sunday, 2 represents Monday, 3 represents Tuesday, and so on. For example, you can find out on what day of the week a holiday occurs.

---

**Important** To avoid errors when using dates, always use four-digit years. For more information about how FileMaker Pro handles two-digit dates, see [Conversion of dates with two-digit years](#).

---

### Examples

DayOfWeek ("10/8/2014") returns **4**.

DayOfWeek (Date (10;9;2014)) returns **5**.

DayOfWeek (ProjectDue) returns **3** when the date in ProjectDue is 10/7/2014.

## DayOfYear

### Purpose

Returns the number of days from the beginning of the year of a date.

### Format

DayOfYear (date)

### Parameters

date - any calendar date

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

DayOfYear (Billing Date) returns **32**, when Billing Date is 2/1/2014.

The following formulas return the total number of days in the current year:

```
DayOfYear (Date (12;31;Year (Get (CurrentDate) ) ) )
```

```
DayOfYear (Date (1;1;Year (Get (CurrentDate) ) + 1) -1)
```

---

**Important** To avoid errors when using dates, always use four-digit years. For more information about how FileMaker Pro handles two-digit dates, see [Conversion of dates with two-digit years](#).

---

## Month

### Purpose

Returns a number from 1 to 12, representing the month of the year in which a date occurs.

### Format

Month(date)

### Parameters

date - any calendar date

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

Month("3/19/2014") returns **3**. This example assumes that the operating system date format is set to MM/DD/YYYY.

Month(Payment) returns **3**, where Payment contains March 19, 2014. (The Payment field must be of type date.)

"Bill Due by: " & Date(Month(DateSold) + 1;Day(DateSold);Year(DateSold)) returns **Bill Due by:** followed by a value that is one month later than DateSold.

---

**Important** To avoid errors when using dates, always use four-digit years. For more information about how FileMaker Pro handles two-digit dates, see Conversion of dates with two-digit years.

---

## MonthName

### Purpose

Returns the full name of the month for a date.

### Format

MonthName (date)

### Parameters

date - any calendar date

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

MonthName ("6/6/2014") returns **June**.

"Payment due by the end of: " & MonthName(Date(Month(InvoiceDate) + 1;Day(InvoiceDate);Year(InvoiceDate))) returns **Payment due by the end of May**, where InvoiceDate is 4/4/2014.

"Payment for: " & MonthName(Date(Month(Payment) + 1;Day(Payment);Year(Payment))) returns **Payment for:** followed by the name of the month that is one past the month of the last payment.

---

**Important** To avoid errors when using dates, always use four-digit years. For more information about how FileMaker Pro handles two-digit dates, see [Conversion of dates with two-digit years](#).

---



## MonthNameJ

### Purpose

Returns the name of the month of a date in Japanese.

### Format

MonthNameJ (date)

### Parameters

date - any calendar date

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

MonthNameJ ("6/6/2014") returns 6月

---

**Important** To avoid errors when using dates, always use four-digit years. For more information about how FileMaker Pro handles two-digit dates, see [Conversion of dates with two-digit years](#).

---

## WeekOfYear

### Purpose

Returns the number of weeks after January 1 of the year of a date.

### Format

WeekOfYear (date)

### Parameters

date - any calendar date

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Fractions of weeks occurring at the beginning or end of the year count as full weeks, so the WeekOfYear function returns values 1 to 54.

---

**Important** To avoid errors when using dates, always use four-digit years. For more information about how FileMaker Pro handles two-digit dates, see Conversion of dates with two-digit years.

---

### Examples

WeekOfYear ("1/1/2014") returns **1**.

WeekOfYear (ProjectDue) returns **6**, when ProjectDue is 2/2/2014.

WeekOfYear ("1/1/2014") - WeekOfYear ("2/2/2014") returns **-5**.

## WeekOfYearFiscal

### Purpose

Returns a number from 1 to 53, representing the week containing a date, from a starting day.

### Format

`WeekOfYearFiscal (date;startingDay)`

### Parameters

`date` - any calendar date

`startingDay` - any number between 1 and 7, where 1 represents Sunday

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

`startingDay` indicates which day is considered the first day of the week.

The first week of the year is the first week that contains four or more days of that year. For example, if you select 1 (Sunday) as the starting day, then January 1 must be on Sunday, Monday, Tuesday, or Wednesday for that week to be the first week of the fiscal year. If you select 2 (Monday) as the starting day, then January 1 must be on Monday, Tuesday, Wednesday, or Thursday for that week to be the first week of the fiscal year.

It is possible, using this function, that dates in a particular year will be returned as the 53rd week of the previous year. For example, if in 2008 you selected Sunday (1) as the starting date, then January 1, 2, or 3 in 2009 would occur in week 53 of fiscal year 2008 (in 2009, January 1 is on a Thursday). The first day of fiscal year 2009 would be on Sunday, January 4, because you selected Sunday (1) as the starting day.

---

**Important** To avoid errors when using dates, always use four-digit years. For more information about how FileMaker Pro handles two-digit dates, see [Conversion of dates with two-digit years](#).

---

### Examples

`WeekOfYearFiscal (Date (1;7;2008) ;1)` returns **2**.

`WeekOfYearFiscal (Date (1;1;2009) ;5)` returns **1**.

`WeekOfYearFiscal (Date (1;2;2009) ;1)` returns **53**.

## Year

### Purpose

Returns a number representing the year in which a date occurs.

### Format

Year (date)

### Parameters

date - any calendar date

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

You can, for example, extract the year from a field containing the date an item was sold.

---

**Important** To avoid errors when using dates, always use four-digit years. For more information about how FileMaker Pro handles two-digit dates, see [Conversion of dates with two-digit years](#).

---

### Examples

Year (DateSold) returns the year stored in DateSold.

Year ("5/5/2014") returns **2014**.

Year (Date (Month (Get (CurrentDate)) + 48 ; Day (Get (CurrentDate)) ; Year (Get (CurrentDate)))) returns the year that is 48 months from today's date.

## YearName

### Purpose

Returns the Japanese year name of a date, in the specified format.

### Format

`YearName (date ; format)`

### Parameters

`date` - any calendar date

`format` - a number (0, 1, or 2) that describes the display format

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

If the value for `format` is blank or other than 0, 1, or 2, then 0 is used.

0 - 明治 8 (Meiji 8), 大正 8 (Taisho 8), 昭和 8 (Showa 8), 平成 8 (Heisei 8), 西暦xxxx (Seireki xxxx [before 1868.9.8])

1 - 明 8 (Mei 8), 大 8 (Tai 8), 昭 8 (Sho 8), 平 8 (Hei 8), (西) xxxx (Sei xxxx [before 1868.9.8])

2 - M8, T8, S8, H8, A.D.xxxx (before 1868.9.8)

Name of Emperor in 0 = Long, 1 = Abbreviated, 2 = 2 byte Roman. Seireki is returned when date is before listed emperors.

### Examples

`YearName (DateField;0)` Returns 平成20 when DateField contains 7/15/2008.

## Design functions

Design [functions](#) return information about the structure of open [database files](#). For example, you could determine the names of all the [layouts](#) or [fields](#) in an open database file.

Design function parameters can be any of the following:

- filenames such as "Customer" or literal text such as "Jack"
- layouts such as `layoutName`
- other functions such as `Left (text ; number)`

### Notes

- FileMaker Pro limits the information returned by a design function, according to the privilege set in effect when the function evaluates a database file. See [Creating and editing privilege sets](#) for more information about granting access to database files.
- Literal text parameters such as filenames and layout names must be enclosed in quotation marks. Use quotation marks around field names to indicate the literal string is the parameter (omit quotation marks to indicate the value stored in the field is the parameter). You can use spaces before or after the parentheses that enclose parameters, but spaces are not necessary. Use a semicolon between parameters when a function requires more than one parameter.
- If you specify a filename as a parameter and the filename contains a period, include the filename extension in the parameter. Otherwise, functions may interpret the period in the filename as the beginning of the filename extension, which can lead to unexpected results.

Click a function name for details.

| This function                      | Returns  |
|------------------------------------|--|
| <a href="#">DatabaseNames</a>      | A list of the names of all files open on the computer.   |
| <a href="#">FieldBounds</a>        | The location, in points, of each field boundary and the field's rotation in degrees.                         |
| <a href="#">FieldComment</a>       | The specified field's comment.   |
| <a href="#">FieldIDs</a>           | A list of all field IDs in <code>fileName</code> and <code>layoutName</code> .                               |
| <a href="#">FieldNames</a>         | A list of the names of all fields on <code>layoutName</code> in <code>fileName</code> .                      |
| <a href="#">FieldRepetitions</a>   | The number and orientation of repetitions of a repeating field as formatted on a layout.                     |
| <a href="#">FieldStyle</a>         | The field formatting applied to <code>fieldName</code> on <code>layoutName</code> in <code>fileName</code> . |
| <a href="#">FieldType</a>          | Information about <code>fieldName</code> .   |
| <a href="#">GetNextSerialValue</a> | The next <a href="#">serial number</a> of <code>fieldName</code> in <code>fileName</code> .                  |
| <a href="#">LayoutIDs</a>          | A list of all layout IDs in <code>fileName</code> .  |
| <a href="#">LayoutNames</a>        | A list of the names of all layouts in <code>fileName</code> .  |
| <a href="#">LayoutObjectNames</a>  | A list of the names of all named objects on <code>layoutName</code> in <code>fileName</code> .               |
| <a href="#">RelationInfo</a>       | A list of four values for each <a href="#">relationship</a> directly related to <code>tableName</code> .     |
| <a href="#">ScriptIDs</a>          | A list of all script IDs in <code>fileName</code> .  |
| <a href="#">ScriptNames</a>        | A list of the names of all <a href="#">scripts</a> in <code>fileName</code> .                                |
| <a href="#">TableIDs</a>           | A list of all <a href="#">table</a> IDs in <code>fileName</code> .   |
| <a href="#">TableNames</a>         | A list of all table occurrences in the relationships graph for <code>fileName</code> .                       |

| <b>This function</b>           | <b>Returns</b>  |
|--------------------------------|---|
| <a href="#">ValueListIDs</a>   | A list of all <a href="#">value list</a> IDs in fileName. |
| <a href="#">ValueListItems</a> | A list of the values in a value list.                     |
| <a href="#">ValueListNames</a> | A list of the names of all value lists in fileName.       |
| <a href="#">WindowNames</a>    | A list of the names of windows that are currently open.   |

## DatabaseNames

### Purpose

Returns a list of the names of all files open on the computer.

### Format

DatabaseNames

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Listed items are separated by carriage returns. The names returned do not include file extensions.

**Note** If your database is hosted on another computer, DatabaseNames returns a list of the names of local [client](#) and remote database files open only on the client computer.

### Examples

To determine whether Customers is one of the files currently open, use the DatabaseNames function with the FilterValues function in the formula:

```
FilterValues ( DatabaseNames ; "Customers" )
```

If the formula returns any text value, then Customers is open.

If you want to know how many files with the same name are open, use the DatabaseNames function with the PatternCount function in the formula:

```
PatternCount ( FilterValues ( DatabaseNames ; "Customers" ) ; "Customers" )
```

This will tell you how many files named Customers are open.



## FieldBounds

### Purpose

Returns the location, in points, of each field boundary and the field's rotation in degrees.

### Format

```
FieldBounds (fileName;layoutName;fieldName)
```

### Parameters

fileName - the name of an open [database file](#) (local or remote).

layoutName - the name of a [layout](#) in the specified database file.

fieldName - the name of a [field](#) on the specified layout.

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

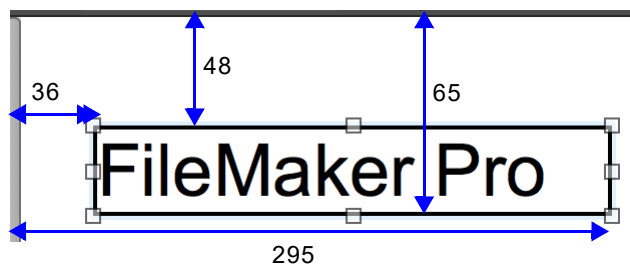
### Description

The location returned is measured from the top-left corner of the layout (regardless of printer margins) and is specified in this order: position of left [field boundary](#), position of top field boundary, position of right field boundary, position of bottom field boundary, degree of rotation (measured in a counter-clockwise direction; 0 degrees for unrotated).

**Note** Your layout begins where your margins end. Because field boundaries are measured from the left side and top of the layout, boundaries returned by `FieldBounds` never change unless you move or re-size a field.

### Examples

`FieldBounds ("Customers"; "Layout #1"; "Field")` returns **36 48 295 65 0** in the example below. Notice that all parameters are enclosed in quotation marks.



## FieldComment

### Purpose

Returns the specified field's comment.

### Format

```
FieldComment (fileName;fieldName)
```

### Parameters

fileName - the name of an open [database file](#) (local or remote).

fieldName - the name of a field in the specified database file.

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

The field name must be in the form `tablename::fieldname` to specify a field that exists in a table different from the current table.

### Examples

`FieldComment("Customers"; "Phone Number")` returns **“Customer's home telephone number”** if it was entered as a comment for the Phone Number field.

`FieldComment("Customers"; "Accounts::Current Balance")` returns **“Customer's current balance”** if it was entered as a comment for the Current Balance field in the Accounts table.

## FieldIDs

### Purpose

Returns a list of all field IDs in fileName and layoutName.

### Format

```
FieldIDs (fileName;layoutName)
```

### Parameters

fileName - the name of an open [database file](#) (local or remote).

layoutName - the name of a [layout](#) or [table](#) in the specified database file.

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Listed items are separated by carriage returns.

[Related fields](#) are returned as **TableID::RelatedFieldID**.

For example, **12::4**, where **12** is the ID of the [table](#) and **4** is the ID of the related field.

If no parameter is specified for fileName, FileMaker returns results for the current file.

### Examples

FieldIDs ("Customers";" ") returns IDs of all unique fields in the default table of Customers.

FieldIDs ("Customers";"Layout#5") returns IDs of all unique fields, including related fields, on Layout#5 in Customers.

## FieldNames

### Purpose

Returns a list of the names of all fields on layoutName in fileName.

### Format

FieldNames (fileName;layoutName)

### Parameters

fileName - the name of an open [database file](#) (local or remote).

layoutName - the name of a [layout](#) or [table](#) in the specified database file.

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Listed items are separated by carriage returns.

Related fields are displayed in tablename::fieldname format.

If no parameter is specified for fileName, FileMaker returns results for the current file.

**Note** If FieldNames returns a question mark (?) or the name of only one field, go to the Specify Calculation dialog box and make sure **Calculation result is** text. Also, you can increase the size of the field on the layout to show more field names.

### Examples

FieldNames ("Customers"; "") returns a list of all the fields in the default table of the Customers database file.

FieldNames ("Customers"; "Data Entry") returns a list of all the fields, including related fields, in the Customers database file that appear on the Data Entry layout.

## FieldRepetitions

### Purpose

Returns the number and orientation of repetitions of a repeating field as formatted on a layout.

### Format

FieldRepetitions (fileName;layoutName;fieldName)

### Parameters

fileName - the name of an open [database file](#) (local or remote).

layoutName - the name of a [layout](#) in the specified file.

fieldName - the name of a field on the specified layout.

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

This function returns the number of repetitions of a repeating field as it is currently formatted on a layout, and the orientation of the field repetitions (horizontal or vertical).

The number of repetitions of fieldName as it is currently formatted could be different from the number of repetitions when the field was defined. If fieldName isn't a [repeating field](#), it returns **1 vertical**.

### Examples

FieldRepetitions("Customers";"Data Entry";"Business Phone") returns **3 vertical** if the Business Phone field is defined as a repeating field with five repetitions but is formatted to only show three repetitions in a vertical orientation on the Data Entry layout.

## FieldStyle

### Purpose

Returns the field formatting applied to fieldName on layoutName in fileName.

### Format

```
FieldStyle(fileName;layoutName;fieldName)
```

### Parameters

fileName - the name of an open [database file](#) (local or remote).

layoutName - the name of a [layout](#) in the specified database file.

fieldName - the name of a [field](#) on the specified layout.

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

If the field has a value list associated with it, the FieldStyle function also returns the name of the [value list](#).

- A standard field returns **Standard**.
- A standard field with a vertical scroll bar returns **Scrolling**.
- A drop-down list returns **Populist**.
- A pop-up menu returns **Popupmenu**.
- A checkbox returns **Checkbox**.
- A radio button returns **RadioButton**.
- A drop-down calendar returns **Calendar**.

### Examples

On the Data Entry layout in the Customers database file, FieldStyle("Customers";"Data Entry";"Current Customer") returns **RadioButton Yes/No List** when the Current Customer field is formatted as a radio button and is associated with the value list named Yes/No List.

## FieldType

### Purpose

Returns information about fieldName.

### Format

FieldType (fileName;fieldName)

### Parameters

fileName - the name of an open [database file](#) (local or remote).

fieldName - the name of a [field](#) in the specified database file.

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Field names must be in the format `tablename::fieldname` to specify a field that exists in a table different from the current table. The result has four values separated by spaces:

- The first value is either Standard, StoredCalc, Summary, UnstoredCalc, External(Secure), External(Open), or Global.
- The second value is the [field type](#): text, number, date, time, timestamp, or [container](#).
- The third value is Indexed or Unindexed.
- The fourth value is the maximum number of repetitions defined for the field (if the field isn't defined as a [repeating field](#), this value is 1).

### Examples

FieldType ("Customers"; "Phone Number") returns **Standard Text Unindexed 3** when, in the Customers database file, the Phone Number field is defined as a text field that repeats a maximum of three times and the storage options are left unchanged. (Most fields are [indexed](#) when a find is performed in that field.)

FieldType ("Customers"; "Current Balance") returns **StoredCalc Number Indexed 1** when, in the Customers database file, the Current Balance field is defined as a stored, numeric calculation field that is indexed.

FieldType ("Customers"; "Today's Date") returns **Global Date Unindexed 1** when, in the Customers database file, the Today's Date field is defined as a [global field](#) of type date. Global fields are never indexed.

FieldType ("Customers"; "Statement") returns **External(Secure) Container Unindexed 1** when, in the Customers database file, the Statement field is defined as a [container field](#) that stores data externally using secure storage. Container fields cannot be indexed.

## GetNextSerialValue

### Purpose

Returns the next serial number of `fieldName` in `fileName`.

### Format

```
GetNextSerialValue (fileName;fieldName)
```

### Parameters

`fileName` - the name of an open [database file](#) (local or remote).

`fieldName` - the name of the [field](#) whose next [serial number](#) you want to determine.

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Field names must be fully qualified in the format `tablename::fieldname` to specify a field that exists in a table different from the current table.

### Examples

`GetNextSerialValue ("Customers"; "CustID")` returns the next serial number for the CustID field.



## LayoutIDs

### Purpose

Returns a list of all layout IDs in fileName.

### Format

LayoutIDs (fileName)

### Parameters

fileName - the name of an open [database file](#) (local or remote).

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Listed items are separated by carriage returns.

If no parameter is specified for fileName, FileMaker returns results for the current file.

### Examples

LayoutIDs ("Customers") returns a list of all the [layout](#) IDs in the Customers database file.

## LayoutNames

### Purpose

Returns a list of the names of all layouts in fileName.

### Format

LayoutNames (fileName)

### Parameters

fileName - the name of an open [database file](#) (local or remote).

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Description

Listed items are separated by carriage returns.

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

LayoutNames ("Customers") returns a list of all the [layouts](#) in the Customers database file.

## LayoutObjectNames

### Purpose

Returns a list of the names of all named objects on layoutName in fileName.

### Format

```
LayoutObjectNames (fileName ; layoutName)
```

### Parameters

fileName - the name of an open [database file](#) (local or remote).

layoutName - the name of a [layout](#) in the specified database file.

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 8.5

### Description

Listed items are separated by carriage returns. Layout objects without object names are not returned. If layoutName isn't specified, then no object names are returned.

Named tab controls, grouped objects, and portal objects that contain other named objects are followed by a list of those named objects enclosed in angle brackets (<>). The angle brackets are shown even if there are no named objects contained within the named tab controls, grouped objects, or portal objects.

### Examples

LayoutObjectNames ("Customers"; "Data Entry") returns a list of named objects in the Customers database file that appear on the Data Entry layout.

## RelationInfo

### Purpose

Returns a list of four values for each relationship directly related to `tableName`.

### Format

```
RelationInfo (fileName ; tableName)
```

### Parameters

`fileName` - the name of an open [database file](#) (local or remote).

`tableName` - the name of a [table](#) in the specified database file.

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Values in a list are separated by carriage returns, and lists are separated by two carriage returns. For each additional [relationship](#) connected to `tableName`, an additional list of four values is output.

The four values are:

- **Source**: Data Source Name of the database table connected to `tableName`.
- **Table**: the name of the table connected to `tableName`.
- **Options**: the options that were set in the right side of the Edit Relationship dialog box when the relationship was defined. This line is blank if the following options are not set; otherwise these options are separated by spaces.
  - **Delete**, if **Delete related records in this table when a record is deleted in the other table** is selected in the right side of the Edit Relationship dialog box.
  - **Create**, if **Allow creation of records in this table via this relationship** is selected in the right side of the Edit Relationship dialog box.
  - **Sorted**, if **Sort records** is selected in the right side of the Edit Relationship dialog box.
- **Relationships**: a list of the defined relationships, one per line. Field names are fully qualified, for example, `TableName::Field Name`.

## Examples

A database file called Human Resources has three tables: Company, Employees, and Addresses. Company::Company ID is connected to Employees::Company ID, Employees::Employee ID is connected to Addresses::Employee ID and Employees::DateOfHire is connected to Addresses::DateMovedIn.

The relationships have the following criteria:

- You can create records in all tables.
- You cannot delete records in all tables.
- A sort was specified for the Addresses table for the Employees<-->Addresses relationship.

RelationInfo("Human Resources"; "Employees") returns:

**Source: Human Resources**

**Table: Company**

**Options: Create**

**Company::Company ID = Employees::Company ID**

**Source: Human Resources**

**Table: Addresses**

**Options: Create Sorted**

**Addresses::Employee ID = Employees::Employee ID**

**Addresses::DateMovedIn >= Employees::DateOfHire**

## ScriptIDs

### Purpose

Returns a list of all script IDs in fileName.

### Format

ScriptIDs(fileName)

### Parameters

fileName - the name of an open [database file](#) (local or remote).

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Listed items are separated by carriage returns.

### Examples

ScriptIDs("Customers") returns a list of all the script IDs in the Customers database file.

## ScriptNames

### Purpose

Returns a list of the names of all scripts in fileName.

### Format

ScriptNames (fileName)

### Parameters

fileName - the name of an open [database file](#) (local or remote).

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Listed items are separated by carriage returns.

If no parameter is specified for fileName, FileMaker returns results for the current file.

### Examples

ScriptNames ("Customers") returns a list of all the scripts in the Customers database file.

## TableIDs

### Purpose

Returns a list of all table IDs in fileName.

### Format

TableIDs(fileName)

### Parameters

fileName - the name of an open [database file](#) (local or remote).

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Listed items are separated by carriage returns.

Each table ID is unique. Also, the ID is independent of when you create each table: the first table could have the smallest, middle, or largest value.

If no parameter is specified for fileName, FileMaker returns results for the current file.

### Examples

TableIDs("University Database") returns

**1065089**

**1065090**

for the University Database database file if two tables have been defined for the file.



## TableNames

### Purpose

Returns a list of all table occurrences in the relationships graph for fileName.

### Format

TableNames (fileName)

### Parameters

fileName - the name of an open [database file](#) (local or remote).

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Listed items are separated by carriage returns.

If no parameter is specified for fileName, FileMaker returns results for the current file.

### Examples

TableNames ("University Database") returns table occurrences

#### Teachers

#### Coaches

for the University Database database file if a Teachers table and a Coaches table have been defined for the file.

## ValueListIDs

### Purpose

Returns a list of all value list IDs in fileName.

### Format

ValueListIDs (fileName)

### Parameters

fileName - the name of an open [database file](#) (local or remote).

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Listed items are separated by carriage returns.

If no parameter is specified for fileName, FileMaker returns results for the current file.

### Examples

ValueListIDs ("Customers") returns a list of all the [value list](#) IDs in the Customers database file.

## ValueListItems

### Purpose

Returns a list of the values in a value list.

### Format

```
ValueListItems (fileName;valuelist)
```

### Parameters

fileName - the name of an open [database file](#), (local or remote).

valuelist - the name of a [value list](#) in the specified database file.

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Listed items are separated by carriage returns.

### Examples

`ValueListItems ("Customers" ; "Code" )` returns a list of all the items in the Code value list in the Customers database file.

## ValueListNames

### Purpose

Returns a list of the names of all value lists in `fileName`.

### Format

`ValueListNames (fileName)`

### Parameters

`fileName` - the name of an open [database file](#) (local or remote).

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Listed items are separated by carriage returns.

If no parameter is specified for `fileName`, FileMaker returns results for the current file.

### Examples

`ValueListNames ("Customers")` returns a list of all the [value list](#) names in the Customers database file.

## WindowNames

### Purpose

Returns a list of the names of windows that are currently open.

### Format

```
WindowNames { (fileName) }
```

### Parameters

{ fileName } - the name of an open [database file](#) (local or remote).

Parameters in braces { } are optional.

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Use the optional `fileName` parameter to only return windows that are based on the specified file. The window could be visible, hidden, or minimized. The order of the names in the list matches the current stacking order of the windows. The visible windows are listed first, then the minimized windows, and then the hidden windows. If there are no databases or windows open, an empty string is returned.

**Note** Even if you close a file, it may remain open as a hidden file if the window of any other file is displaying data from that file. (For example, another window may be displaying related data from the file you attempted to close.) FileMaker Pro will close the file when you close all the dependent windows.

### Examples

`WindowNames` returns **Customers** and **Invoices** separated by a carriage return when those windows are currently open.

`WindowNames ("contacts")` returns a list of windows that are based on the contacts database file.

## External functions

Use external [functions](#) to access FileMaker Pro [plug-ins](#). Plug-ins add features to FileMaker Pro. For more information, see Setting plug-in preferences.

External functions are only available if FileMaker Pro plug-ins are installed and enabled on your computer. If no FileMaker Pro plug-ins are installed, you see only the generic external function definition in the Specify Calculation dialog box:

External (nameOfFunction; parameter)

### Plug-ins written for version 7.0 and later

Each plug-in defines its own functions and parameters. See the documentation that came with the plug-in for each function's usage.

### Plug-ins written for version 6.0 and earlier

These plug-ins are still supported and continue to use the External function to access the plug-in's functions. The first parameter is the name of the plug-in function to execute and the second is a parameter that is passed to that function. See the documentation that came with the plug-in for each function's usage.

| This function            | Does this   |
|--------------------------|---|
| <a href="#">External</a> | Accesses plug-ins created for versions of FileMaker Pro prior to 7.0, and uses the syntax External("function name", parameter). |

For more information, see Updating plug-ins.

## External

### Purpose

Accesses plug-ins created for versions of FileMaker Pro prior to 7.0, and uses the syntax `External("function name", parameter)`.

### Format

`External (nameOfFunction;parameter)`

### Parameters

`nameOfFunction` - the name of the external function

`parameter` - the parameter(s) required by the external function. A parameter is required, even if it's only 0.

### Data type returned

Depends on the external function

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Plug-ins created for FileMaker Pro version 7.0 and later do not use the `External ("function name", parameter)` syntax. For more information, see [External functions](#).

## Financial functions

Financial [functions](#) calculate financial information, such as net present value and payments. For example, you can calculate the monthly payments required to buy a car at a certain loan rate using the [PMT](#) function.

Click a function name for details.

| This function       | Returns   |
|---------------------|---|
| <a href="#">FV</a>  | The future value (FV) of an initial investment, based on a constant interest rate and payment amount for the number of periods in months. |
| <a href="#">NPV</a> | The net present value (NPV) of a series of unequal payments made at regular intervals, assuming a fixed interest rate per interval.       |
| <a href="#">PMT</a> | The payment (PMT) required by the term, interest rate, and principal.   |
| <a href="#">PV</a>  | The present value (PV) of a series of equal payments made at regular intervals, assuming a fixed interest rate per interval.              |



## FV

### Purpose

Returns the future value (FV) of an initial investment, based on a constant interest rate and payment amount for the number of periods in months.

### Format

`FV(payment;interestRate;periods)`

### Parameters

`payment` - payment to be made per period

`interestRate` - interest rate per period

`periods` - number of periods

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Use this function to calculate `FV`. For example, you can calculate how much you'll earn on an investment in which you pay \$50 a month for 60 months at a 6 percent annual interest rate.

### Notes

- When `interestRate` is 0, this function returns the result of `payment * periods`.
- The `FV` function doesn't account for the present value of your investment, and it assumes that payment is made at the end of each period.

$$FV = \text{payment} * \frac{(1 + \text{interestRate})^{\text{periods}} - 1}{\text{interestRate}}$$

### Examples

`FV(50;.11/12;5 * 12)` returns **3975.90398429....**

`FV(2000;.12;30) + 5000 * (.12 + 1) ^ 30` returns **632464.97928640....**

`FV(500;.11/5;60)` returns **61141.65130790....**

To set the decimal precision of the returned value, enclose the current formulas with the `Round` function. For example, `Round(Current Formula;2)`.

## NPV

### Purpose

Returns the net present value (NPV) of a series of unequal payments made at regular intervals, assuming a fixed interest rate per interval.

### Format

`NPV(payment;interestRate)`

### Parameters

`payment` - a [repeating field](#) containing unequal payment amounts, or an [expression](#) that returns a reference to one.

`interestRate` - interest rate.

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Use this function to calculate NPV. For example, if someone borrows money from you and pays you back in unequal amounts over a period of several years, you can use the NPV function to calculate the result.

$$\text{NPV} = \frac{\text{loan amount}}{1 + \text{interestRate}} + \frac{\text{first payment}}{(1 + \text{interestRate})^2} + \frac{\text{second payment}}{(1 + \text{interestRate})^3} + \dots + \frac{\text{n}^{\text{th}} \text{ payment}}{(1 + \text{interestRate})^{n+1}}$$

### Examples

`NPV(Loan;.05)` returns **156.91277445...**, when the repeating field, Loan, contains -2000 (the initial payment), 600, 300, 500, 700, and 400. The result (156.91277445...) is the actual profit in today's dollars that will be realized from this transaction.

`NPV(Amounts;.10)` returns **16758.35604870...**, when the repeating field, Amounts, contains -5000 (the initial investment), 10,000, 0, 10,000, and 10,000.

If you want each return value to return 2 decimal places, surround the current formulas with the correct Round function: `Round(Current Formula;2)`.

## PMT

### Purpose

Returns the payment (PMT) required by the term, interest rate, and principal.

### Format

`PMT(principal;interestRate;term)`

### Parameters

`principal` - principal amount.

`interestRate` - interest rate. If the interest rate is annual, divide the rate by 12.

`term` - length of time, expressed in number of months.

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Use this function to calculate `PMT`.

$$\text{PMT} = \text{payment} \left( \frac{1 - (1 + \text{interestRate})^{-\text{periods}}}{\text{interestRate}} \right)$$

### Examples

In the following example, the `PMT` function calculates payments for purchasing a sports car costing \$21,000, at an annual rate of 6.9% over 48 monthly payments.

`PMT(21000;.069/12;48)` returns the payment amount **\$501.90**.

`PMT(Cost;.13;Years)` returns a payment amount, based on the purchase value stored in `Cost`, at a 13 percent rate, over the duration stored in `Years`.

"Your payment will be " & `PMT(150000;.13/12;Months)` & "." returns **Your payment will be**, followed by the payment amount, based on a total cost of \$150,000, at a 13 percent annual percentage rate, over the duration stored in `Months`.

## PV

### Purpose

Returns the present value (PV) of a series of equal payments made at regular intervals, assuming a fixed interest rate per interval.

### Format

PV(payment;interestRate;periods)

### Parameters

payment - payment amount to be made per period. Type a negative number for money you pay and a positive number for money you receive.

interestRate - interest rate per period.

periods - number of periods (intervals between payments).

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Use this function to calculate PV.

$$PV = \text{payment} * \frac{1 - (1 + \text{interestRate})^{-\text{periods}}}{\text{interestRate}}$$

**Note** When interestRate is 0, this function returns the result of payment \* periods.

### Examples

Your cousin borrowed \$2,000 from you, offering to pay you back \$500 a year for five years, for a total of \$2,500 at the end of five years. If inflation was 5 percent annually, with the following entry you could find out what those payments are worth with the PV function.

PV(500;.05;5) returns **2164.73833531....**

If you want the return value to return two decimal places, enclose the formula with the correct Round function: Round(Current Formula;2).

## Get functions

Get functions monitor errors in scripts, or capture information about file status or actions being performed.

Many Get [functions](#) return information that changes on a regular basis. For example, when the Get(CurrentTime) function is placed in a stored [calculation field](#), the time will update only when a new [record](#) is created. If the calculation has other [fields](#) in it but the calculation result still returns the current time, the stored calculation result will update only when those other fields have been modified in the current record. If either of these calculations are [unstored](#), the time will update as needed. For performance reasons, making a calculation field unstored is not always the best idea. Get functions are best used in a script where the status information from a Get function is up-to-date at the moment that the calculation is run.

Click a function name for details.

| This function                                    | Returns   |
|--|---|
| <a href="#">Get(AccountExtendedPrivileges)</a>   | A list of keywords, based on the account used to open the file, for the enabled <a href="#">extended privileges</a> .             |
| <a href="#">Get(AccountName)</a>                 | The authenticated account name being used by the file's current user.   |
| <a href="#">Get(AccountPrivilegeSetName)</a>     | The <a href="#">privilege set</a> name being used by the account used to open the file.   |
| <a href="#">Get(ActiveFieldContents)</a>         | The contents of the field that has the <a href="#">focus</a> .  |
| <a href="#">Get(ActiveFieldName)</a>             | The name of the field that has the focus.   |
| <a href="#">Get(ActiveFieldTableName)</a>        | The name of the <a href="#">table</a> that contains the field that has the focus.   |
| <a href="#">Get(ActiveLayoutObjectName)</a>      | The name of the layout object that has the focus in the calculation's current window.   |
| <a href="#">Get(ActiveModifierKeys)</a>          | A number for the keyboard modifier keys being pressed.  |
| <a href="#">Get(ActivePortalRowNumber)</a>       | The number of the <a href="#">portal</a> row that has the focus.  |
| <a href="#">Get(ActiveRepetitionNumber)</a>      | A number representing the repetition that has the focus in a <a href="#">repeating field</a> .                                    |
| <a href="#">Get(ActiveSelectionSize)</a>         | A number representing how many characters are selected.   |
| <a href="#">Get(ActiveSelectionStart)</a>        | A number representing the starting character of the selected text.  |
| <a href="#">Get(AllowAbortState)</a>             | 1 (true) if the Allow User Abort script step is on; otherwise, returns 0 (false).   |
| <a href="#">Get(AllowFormattingBarState)</a>     | 1 (true) if the formatting bar is allowed to be visible; otherwise, returns 0 (false).  |
| <a href="#">Get(ApplicationArchitecture)</a>     | The current application architecture.   |
| <a href="#">Get(ApplicationLanguage)</a>         | Text representing the current application language.   |
| <a href="#">Get(ApplicationVersion)</a>          | Text representing the FileMaker product and version.  |
| <a href="#">Get(CalculationRepetitionNumber)</a> | A number representing the repetition of the calculation field being calculated.   |
| <a href="#">Get(ConnectionAttributes)</a>        | The name of the current file's host, and the certificate authority that issued the SSL certificate used to secure the connection. |
| <a href="#">Get(ConnectionState)</a>             | A number representing the security state of the network connection for the current file.  |

| This function                                   | Returns  |
|---|--|
| <a href="#">Get(CurrentDate)</a>                | The current date according to the system calendar.   |
| <a href="#">Get(CurrentExtendedPrivileges)</a>  | A list of keywords, based on the account used to evaluate this calculation, for the enabled extended privileges.     |
| <a href="#">Get(CurrentHostTimestamp)</a>       | The host's current date and time (to the nearest second) according to the system clock.                              |
| <a href="#">Get(CurrentPrivilegeSetName)</a>    | The name of the privilege set being used to evaluate this calculation in the file.                                   |
| <a href="#">Get(CurrentTime)</a>                | The current time (to the nearest second) according to the system clock.  |
| <a href="#">Get(CurrentTimestamp)</a>           | The current date and time (to the nearest second) according to the system clock.                                     |
| <a href="#">Get(CurrentTimeUTCMilliseconds)</a> | The current time in Coordinated Universal Time (UTC) to the nearest millisecond.                                     |
| <a href="#">Get(CustomMenuSetName)</a>          | The name of the active custom menu set.  |
| <a href="#">Get(DesktopPath)</a>                | The path to the desktop folder for the current user.   |
| <a href="#">Get(Device)</a>                     | A number representing the type of computer or mobile device running the FileMaker product.                           |
| <a href="#">Get(DocumentsPath)</a>              | The path to the Documents folder for the current user.   |
| <a href="#">Get(DocumentsPathListing)</a>       | A list of all files and folders in the Documents folder returned by the <a href="#">Get(DocumentsPath)</a> function. |
| <a href="#">Get(EncryptionState)</a>            | A value representing the file's current encryption state.  |
| <a href="#">Get(ErrorCaptureState)</a>          | 1 (true) if the Set Error Capture script step is on; otherwise, returns 0 (false).                                   |
| <a href="#">Get(FileMakerPath)</a>              | The path to the folder of the currently running copy of FileMaker Pro.   |
| <a href="#">Get(FileName)</a>                   | The name of the currently active file, without the filename extension.   |
| <a href="#">Get(FilePath)</a>                   | The full path of the location of the currently active file.  |
| <a href="#">Get(FileSize)</a>                   | The size, in bytes, of the currently active file.  |
| <a href="#">Get(FoundCount)</a>                 | The number of records in the current <a href="#">found set</a> .   |
| <a href="#">Get(HighContrastColor)</a>          | The current high contrast default color scheme name.   |
| <a href="#">Get(HighContrastState)</a>          | 1 (true) if high contrast is available and active; otherwise, returns 0 (false).                                     |
| <a href="#">Get(HostApplicationVersion)</a>     | The version of FileMaker Pro or FileMaker Server running on the computer hosting the current file.                   |
| <a href="#">Get(HostIPAddress)</a>              | The IP address of the host computer for the current file.  |
| <a href="#">Get(HostName)</a>                   | The registered name of the computer that is <a href="#">hosting</a> the file.  |
| <a href="#">Get(InstalledFMPlugins)</a>         | The name, version number, and enabled state of installed <a href="#">plug-ins</a> .                                  |
| <a href="#">Get(LastError)</a>                  | A number for the error that occurred in the execution of the most recently executed script step.                     |
| <a href="#">Get(LastExternalErrorDetail)</a>    | Text about errors external to FileMaker that are returned by <a href="#">Get(LastError)</a> .                        |

| This function                                | Returns   |
|--|---|
| <a href="#">Get(LastMessageChoice)</a>       | A number corresponding to the button clicked in an alert message displayed by the Show Custom Dialog script step.   |
| <a href="#">Get(LayoutAccess)</a>            | A number based on record <a href="#">access privileges</a> available through the current layout.                    |
| <a href="#">Get(LayoutCount)</a>             | The total number of <a href="#">layouts</a> in the file.  |
| <a href="#">Get(LayoutName)</a>              | The name of the layout currently being displayed.   |
| <a href="#">Get(LayoutNumber)</a>            | The number of the layout (specified in the Manage Layouts dialog box) currently displayed.                          |
| <a href="#">Get(LayoutTableName)</a>         | The name of the table displaying records in the current layout.   |
| <a href="#">Get(LayoutViewState)</a>         | A number representing the currently active file view.   |
| <a href="#">Get(MenubarState)</a>            | A number for the current menu bar state.  |
| <a href="#">Get(ModifiedFields)</a>          | A list of fields that have been modified in the current record of the current table.                                |
| <a href="#">Get(MultiUserState)</a>          | A number representing the level of sharing for the file using FileMaker Network.                                    |
| <a href="#">Get(NetworkProtocol)</a>         | The name of the <a href="#">network protocol</a> (TCP/IP) that FileMaker is using on this machine.                  |
| <a href="#">Get(NetworkType)</a>             | A number representing the type of network being used to access the current file.                                    |
| <a href="#">Get(PageNumber)</a>              | A number representing the current page being printed or previewed.  |
| <a href="#">Get(PersistentID)</a>            | Text representing a unique identifier of the computer or device running FileMaker.                                  |
| <a href="#">Get(PreferencesPath)</a>         | The path to the preferences and default options folder for the current user.  |
| <a href="#">Get(PrinterName)</a>             | Text identifying the default printer name.  |
| <a href="#">Get(QuickFindText)</a>           | The text that is stored in the quick find box.  |
| <a href="#">Get(RecordAccess)</a>            | A number based on the current record's access privileges, assigned through the Custom Record Privileges dialog box. |
| <a href="#">Get(RecordID)</a>                | The unique ID number of the current record.   |
| <a href="#">Get(RecordModificationCount)</a> | The total number of times changes to the current record have been committed.  |
| <a href="#">Get(RecordNumber)</a>            | The number of the current record in the current found set.  |
| <a href="#">Get(RecordOpenCount)</a>         | The total number of open records in the current found set that haven't been saved.                                  |
| <a href="#">Get(RecordOpenState)</a>         | A number representing the state of the current record.  |
| <a href="#">Get(RequestCount)</a>            | The total number of find requests defined for the current table.  |
| <a href="#">Get(RequestOmitState)</a>        | 1 (true) if Omit is selected in the current find request; otherwise, returns 0 (false).                             |
| <a href="#">Get(ScreenDepth)</a>             | The number of bits needed to represent the color or shade of gray of a pixel on the main screen.                    |
| <a href="#">Get(ScreenHeight)</a>            | The height, in points, of the screen in which the window of the current file is open.                               |

| This function                              | Returns   |
|--|---|
| <a href="#">Get(ScreenScaleFactor)</a>     | The scale factor of the screen in which the current file is open.   |
| <a href="#">Get(ScreenWidth)</a>           | The width, in points, of the screen in which the window of the current file is open.  |
| <a href="#">Get(ScriptAnimationState)</a>  | 1 (true) if animations are enabled for the current script; otherwise, returns 0 (false).                                    |
| <a href="#">Get(ScriptName)</a>            | The name of the script that is currently running or is paused.  |
| <a href="#">Get(ScriptParameter)</a>       | The script parameter passed into the current script.  |
| <a href="#">Get(ScriptResult)</a>          | The script result from a performed sub-script.  |
| <a href="#">Get(SortState)</a>             | A value representing the current sort state.  |
| <a href="#">Get(StatusAreaState)</a>       | A number representing the current <a href="#">status toolbar</a> state.   |
| <a href="#">Get(SystemDrive)</a>           | The drive letter (Windows) or volume name (OS X) where the running operating system is located.                             |
| <a href="#">Get(SystemIPAddress)</a>       | A list of the <a href="#">IP addresses</a> of all computers connected to an active NIC (Network Interface Controller) card. |
| <a href="#">Get(SystemLanguage)</a>        | The language currently set on the current system.   |
| <a href="#">Get(SystemNICAddress)</a>      | The hardware addresses of all NIC (Network Interface Controller) cards connected to the computer.                           |
| <a href="#">Get(SystemPlatform)</a>        | A number representing the current platform.   |
| <a href="#">Get(SystemVersion)</a>         | The version of the current operating system.  |
| <a href="#">Get(TemporaryPath)</a>         | The path to the temporary folder FileMaker Pro uses for the current user, or the path FileMaker Server uses on the system.  |
| <a href="#">Get(TextRulerVisible)</a>      | 1 (true) if the text ruler is displayed; otherwise, returns 0 (false).  |
| <a href="#">Get(TotalRecordCount)</a>      | The total number of records in the current table.   |
| <a href="#">Get(TouchKeyboardState)</a>    | 1 (true) if the touch keyboard is set to display automatically when needed; otherwise, returns 0 (false).                   |
| <a href="#">Get(TriggerCurrentPanel)</a>   | The index and object name of the panel to switch from when the OnPanelSwitch <a href="#">script trigger</a> is activated.   |
| <a href="#">Get(TriggerExternalEvent)</a>  | A number representing the event that activated an OnExternalCommandReceived script trigger.                                 |
| <a href="#">Get(TriggerGestureInfo)</a>    | Details about the gesture that activated an OnGestureTap script trigger.  |
| <a href="#">Get(TriggerKeystroke)</a>      | The characters that activated an OnObjectKeystroke or OnLayoutKeystroke script trigger.                                     |
| <a href="#">Get(TriggerModifierKeys)</a>   | The state of the keyboard modifier keys when an OnObjectKeystroke or OnLayoutKeystroke script trigger was activated.        |
| <a href="#">Get(TriggerTargetPanel)</a>    | The index and the object name of the panel to switch to when the OnPanelSwitch script trigger is activated.                 |
| <a href="#">Get(UserCount)</a>             | The number of clients currently accessing the file.   |
| <a href="#">Get(UserName)</a>              | The name of the FileMaker user, as specified in the General tab of the Preferences dialog box.                              |
| <a href="#">Get(UseSystemFormatsState)</a> | 1 (true) if Use System Formats in the Format menu is on; otherwise, returns 0 (false).                                      |



| This function                            | Returns   |
|--|---|
| <a href="#">Get(UUID)</a>                | Text representing a Universally Unique Identifier (UUID).   |
| <a href="#">Get(WindowContentHeight)</a> | The height, in points, of the window content area.  |
| <a href="#">Get(WindowContentWidth)</a>  | The width, in points, of the window content area.   |
| <a href="#">Get(WindowDesktopHeight)</a> | The height, in points, of the desktop space.  |
| <a href="#">Get(WindowDesktopWidth)</a>  | The width, in points, of the desktop space.   |
| <a href="#">Get(WindowHeight)</a>        | The height, in points, of the window on which the script is acting.   |
| <a href="#">Get(WindowLeft)</a>          | The horizontal distance, in points, of the outer edge of the window on which the script is acting, relative to the leftmost edge of the screen. |
| <a href="#">Get(WindowMode)</a>          | A number representing the mode the FileMaker product is in when the function is evaluated.  |
| <a href="#">Get(WindowName)</a>          | The name of the window on which the script is acting.   |
| <a href="#">Get(WindowOrientation)</a>   | A value representing the orientation of the window on which the script is acting.   |
| <a href="#">Get(WindowStyle)</a>         | The style of the window on which the script is acting.  |
| <a href="#">Get(WindowTop)</a>           | The vertical distance, in points, of the outer edge of the window on which the script is acting, relative to the bottom edge of the menu bar.   |
| <a href="#">Get(WindowVisible)</a>       | 1 (true) if the window is visible; returns 0 (false) if the window is hidden using Hide Window.   |
| <a href="#">Get(WindowWidth)</a>         | The width, in points, of the window on which the script is acting.  |
| <a href="#">Get(WindowZoomLevel)</a>     | The zoom percentage of the current window.  |

## Get functions example

This script uses the function `Get (CurrentDate)` to check each record in the found set to see if an account is past due. If an account is past due, the script shows a message and prompts the user to click a button labeled Ignore, Send Letter, or Send Mail (set up through the Show Custom Dialog script step). The script captures the user's response using `Get (LastMessageChoice)`. Then, based on the user's response, the script performs an action: it cancels the rest of the script, prints a "payment is late" letter, or sends email to the associated account.

```

Go to Layout ["LayoutName"]
Go to Record/Request/Page [First]
Loop
  If [DatabaseName::Date < Get (CurrentDate) - 30]
    Show Custom Dialog ["30 or more days late"]
    If [Get (LastMessageChoice) = 1]
      Halt Script
    Else If [Get (LastMessageChoice) = 2]
      Go to Layout ["Late Notice"]
      Print []
    Else
      Send Mail [Send via E-mail Client; With dialog: Off; To:
DatabaseName::Client; Subject: "Late Notice"; Message: "Your account is
past due."]
      End If
    End If
  Go to Record/Request/Page [Exit after last, Next]
End Loop
Go to Layout [original layout]

```

## Get(AccountExtendedPrivileges)

### Purpose

Returns a list of keywords, based on the account used to open the file, for the enabled extended privileges.

### Format

Get (AccountExtendedPrivileges)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 11.0

### Description

The keywords are separated by carriage returns. [Extended privileges](#) are additional access rights assigned to an account's [privilege set](#). See [About accounts, privilege sets, and extended privileges](#).

Returns an empty list if a user doesn't have extended privileges assigned to the account used to open the file.

### Notes

- If you specify the [context](#) for the current calculation, this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.
- See also [Get\(CurrentExtendedPrivileges\) function](#).

### Example 1

If you are logged in with a privilege set that includes only the Access via FileMaker WebDirect extended privilege:

Get ( AccountExtendedPrivileges ) returns **fmwebdirect**.

### Example 2

If you are logged in and running a script with full access privileges, Get(AccountExtendedPrivileges) returns the extended privileges for your account. (By contrast, Get(CurrentExtendedPrivileges) returns the extended privileges for the Admin account.)

Assume the [Read-Only Access] privilege set includes the Access via FileMaker Network extended privilege, and the [Full Access] privilege set includes the Access via FileMaker Network and Access via FileMaker WebDirect extended privileges. If you are logged in with the Guest account and run a script set to run with full access privileges:

Get ( AccountExtendedPrivileges ) returns **fmapp**

Get ( CurrentExtendedPrivileges ) returns  
**fmapp**  
**fmwebdirect**

## Get(AccountName)

### Purpose

Returns the authenticated account name being used by the file's current user.

### Format

Get (AccountName)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

Use this function for FileMaker [authentication](#). If a user is using the default Admin [account](#), Get (AccountName) returns **Admin**. If a user is using the FileMaker Pro guest account, then **[Guest]** will be returned.

For external server authentication, Get (AccountName) returns the name of the authenticated account being used by the current user of the database file, not the group the user belongs to (the group name appears in the **Account** list when you define accounts and privileges in FileMaker Pro). If an individual belongs to more than one group (account), the first group name listed when you choose **View By Authentication Order** while defining accounts and privileges determines access for the user.

### Notes

- Get(AccountName) returns the account name as it was entered by the user. When a user logs in, the account name is not case sensitive, so Get(AccountName)'s result may not be identical to the account name as it appears in the Manage Security dialog box.
- If you specify the [context](#) for the current calculation, this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.

### Example 1

Returns **Marketing** when Marketing is the name of the account that was used to log in to the database file.

### Example 2

If a user enters **MARKETING** when logging in to the Marketing account, Get(AccountName) returns **MARKETING**.

## Get(AccountPrivilegeSetName)

### Purpose

Returns the privilege set name being used by the account used to open the file.

### Format

Get (AccountPrivilegeSetName)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 11.0

### Description

If a user is using the default Admin account and you haven't modified [access privileges](#) for the file, this function returns **[Full Access]**.

### Notes

- If you specify the [context](#) for the current calculation, this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.
- See also [Get\(CurrentPrivilegeSetName\) function](#).

### Examples

For an Administrator, Get (AccountPrivilegeSetName) might return **[Full Access]**.

For a user in the sales department, Get (AccountPrivilegeSetName) might return **[Data Entry Only]**.

For a user with Read-Only Access to a database who is running a script that is set to run with full access privileges, Get (AccountPrivilegeSetName) returns **[Read-Only Access]** but Get (CurrentPrivilegeSetName) returns **[Full Access]** (for the current script).

## Get(ActiveFieldContents)

### Purpose

Returns the contents of the field that has the focus.

### Format

Get (ActiveFieldContents)

### Parameters

None

### Data type returned

text, number, date, time, timestamp, container

### Originated in

FileMaker Pro 6.0 or earlier

### Description

When the [focus](#) is in a [repeating field](#), this function returns the contents of the active repetition. The result type of the active field depends upon the data type of the active field and the result type assigned to the Get(ActiveFieldContents) calculation function.

### Examples

Returns **SomeShop** when the focus is in the Name field, and that field contains the data SomeShop.

This type of calculation is most useful if used in a script when you want to examine data in different fields as the script proceeds.

## Get(ActiveFieldName)

### Purpose

Returns the name of the field that has the focus.

### Format

Get (ActiveFieldName)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

Returns **Country**, when the focus is in the Country field.

## Get(ActiveFieldName)

### Purpose

Returns the name of the table that contains the field that has the focus.

### Format

Get (ActiveFieldName)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

If there is no active field, this function returns an empty string.

### Examples

There are two fields, Teachers::Name and Coaches::Name, on the current [layout](#). Creating a script that returns the result of `Get (ActiveFieldName)` to a third field will return **Teachers** when the script is performed after clicking in the Teachers::Name field, or will return **Coaches** after clicking in the Coaches::Name field.

## Get(ActiveLayoutObjectName)

### Purpose

Returns the name of the layout object that has the focus in the calculation's current window.

### Format

Get (ActiveLayoutObjectName)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 8.5

### Description

When no layout object has the [focus](#), this function returns an empty string.

### Examples

There is a named field on the current [layout](#) called `customerName`. When the focus is on the field, `Get (ActiveLayoutObjectName)` returns **customerName**.



## Get(ActiveModifierKeys)

### Purpose

Returns a number for the keyboard modifier keys being pressed.

### Format

Get (ActiveModifierKeys)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

The number returned is calculated by summing numbers representing each modifier key (for example, Control+Shift) being pressed. The values assigned to the keys are:

- Shift = 1
- Caps Lock = 2
- Ctrl (Windows) and Control (OS X) = 4
- Alt (Windows) and Option (OS X) = 8
- Command (OS X) = 16

### Notes

- In FileMaker WebDirect, this function is not supported and returns **0**.

### Examples

Returns **9** when Shift+Alt is pressed on a computer running Windows.

You could use this function in a script that includes a custom dialog box script step (with an **OK** and **Cancel** button) to perform some special action if the user presses the Alt (or Option) key while clicking **OK**.

## Get(ActivePortalRowNumber)

### Purpose

Returns the number of the portal row that has the focus.

### Format

Get (ActivePortalRowNumber)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

When no [portal](#) row contains the [focus](#), this function returns 0. If there are multiple windows open in the current file, each window can have its own portal row number value, but results are returned for only the foreground window. If a user navigates to a portal without selecting a specific portal row and without making an object active within a specific portal row, `Get (ActivePortalRowNumber)` returns row 0 rather than row 1.

### Notes

- If a field on a layout is defined as `Get (ActivePortalRowNumber)`, the window must be refreshed before the field will display the current portal row number.

### Examples

Returns **5** when the fifth row of a portal has the focus, or when the focus is in a [field](#) in the fifth portal row.

Returns **1** after the `Go to Portal Row [First]` script step runs.

Returns **0** when a portal is not selected.

## Get(ActiveRepetitionNumber)

### Purpose

Returns a number representing the repetition that has the focus in a repeating field.

### Format

Get (ActiveRepetitionNumber)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

The first repetition returns 1. If the current field isn't a [repeating field](#), this function returns 1.

### Examples

Returns **5** when the [focus](#) is in the fifth repetition of a repeating field.

## Get(ActiveSelectionSize)

### Purpose

Returns a number representing how many characters are selected.

### Format

Get (ActiveSelectionSize)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

Returns 0 if there is no selection.

### Notes

- In FileMaker WebDirect, this function is not supported and returns an empty string.
- In FileMaker Go, this function may return an inaccurate value if the selected text is in a [concealed edit box](#).

### Examples

Returns **4** when 4 characters are selected.

## Get(ActiveSelectionStart)

### Purpose

Returns a number representing the starting character of the selected text.

### Format

Get (ActiveSelectionStart)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

Returns the cursor's current position if no text is selected.

If there are multiple windows open in the current [database file](#), a result is returned for only the foreground window.

### Notes

- In FileMaker WebDirect, Get(ActiveSelectionStart) returns a value only if the selected text is in a field that displays as an edit box.
- In FileMaker Go, this function may return an inaccurate value if the selected text is in a [concealed edit box](#).

### Examples

Returns **5** when the selection starts at character 5.

## Get(AllowAbortState)

### Purpose

Returns 1 (true) if the Allow User Abort script step is on; otherwise, returns 0 (false).

### Format

Get (AllowAbortState)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Examples

Returns **1** if the Allow User Abort script step is on.

## Get(AllowFormattingBarState)

### Purpose

Returns 1 (true) if the formatting bar is allowed to be visible; otherwise, returns 0 (false).

### Format

Get (AllowFormattingBarState)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 8.0

### Description

The Allow Formatting Bar script step sets the formatting bar state.

### Notes

- In FileMaker WebDirect, this function is not supported and returns **0**.

### Examples

Returns **1** if the formatting bar is allowed to be visible.

## Get(ApplicationArchitecture)

### Purpose

Returns the current application architecture.

### Format

Get (ApplicationArchitecture)

### Parameters

None

### Data type returned

Text

### Originated in

FileMaker Pro 14.0

### Description

Returns:

- **i386** for the 32-bit version of FileMaker Pro
- **x86\_64** for the 64-bit version of FileMaker Pro, FileMaker Server, FileMaker WebDirect, and Custom Web Publishing
- **arm7** for FileMaker Go running on an ARMv7-based device
- **arm7s** for FileMaker Go running on an ARMv7s-based device
- **arm64** for FileMaker Go running on a 64-bit ARM-based device

### Example 1

Detects the application architecture, then installs the appropriate version of a plug-in.

```
If [Get ( ApplicationArchitecture ) = "i386"]
    Install Plug-in File [Plugins::32bit]
Else If [Get ( ApplicationArchitecture ) = "x86_64"]
    Install Plug-in File [Plugins::64bit]
End If
```



## Get(ApplicationLanguage)

### Purpose

Returns text representing the current application language.

### Format

Get (ApplicationLanguage)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

The text that is returned by this function is in the English language.

For hosted databases, Get(ApplicationLanguage) returns the client's current language.

**Note** In FileMaker WebDirect, Get(ApplicationLanguage) returns the web browser's current language.

FileMaker Pro supports:

- English
- French
- Italian
- German
- Swedish
- Spanish
- Dutch
- Japanese
- Simplified Chinese
- Brazilian Portuguese
- Korean

### Examples

Returns **English** when the current application language is English.

## Get(ApplicationVersion)

### Purpose

Returns text representing the FileMaker product and version.

### Format

Get (ApplicationVersion)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Returns:

- **Pro version** for FileMaker Pro
- **ProAdvanced version** for FileMaker Pro Advanced
- **Runtime version** for FileMaker Runtime
- **Web Publishing Engine version** for FileMaker Server Web [Client](#)
- **xDBC version** for xDBC Client
- **Server version** for FileMaker Server
- **Go version** for FileMaker Go on iPhone or iPod touch
- **Go\_iPad version** for FileMaker Go on iPad

### Examples

Returns **Pro 15.0.1** in FileMaker Pro 15.0.1. For an example script, see Go to Layout script step.

## Get(CalculationRepetitionNumber)

### Purpose

Returns a number representing the repetition of the calculation field being calculated.

### Format

Get (CalculationRepetitionNumber)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

The first repetition returned is 1. If the current field isn't a [repeating field](#), the function returns 1.

### Example 1

Returns **5** when FileMaker Pro is calculating the fifth repetition of a repeating field.

### Example 2

As the field definition for Timer::Reminder (a calculation field with five repetitions), calculates future times based on the contents of Timer::Start Time (a time field) and Timer::Snooze Minutes (a number field):

```
Time ( Hour ( Extend ( Timer::Start Time ) ) ; Minute ( Extend (
Timer::Start Time ) ) + ( Extend ( Timer::Snooze Minutes ) * Get (
CalculationRepetitionNumber ) ) ; Seconds ( Extend ( Timer::Start Time ) )
)
```

**Note** The [Extend function](#) allows a non-repeating field to be used in every repetition of a repeating field.

When Timer::Start Time contains "12:00:00" and Timer::Snooze Minutes contains "6":

- Timer::Reminder[1] returns **12:06:00**
- Timer::Reminder[2] returns **12:12:00**
- Timer::Reminder[3] returns **12:18:00**
- Timer::Reminder[4] returns **12:24:00**
- Timer::Reminder[5] returns **12:30:00**

## Get(ConnectionAttributes)

### Purpose

Returns the name of the current file's host, and the certificate authority that issued the SSL certificate used to secure the connection.

### Format

Get (ConnectionAttributes)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 13.0

### Description

Get(ConnectionAttributes) returns an empty string if:

- the current file is not hosted
- the host is not FileMaker Server
- the host does not use an SSL certificate to secure the connection to the [client](#)

### Examples

If the [host](#) is named "group\_server" and the current SSL certificate was issued by XYZ Inc., Get(ConnectionAttributes) returns:

**[ Peer Certificate ]**  
**commonName: group\_server**  
**CA Issuers: XYZ Inc.**

## Get(ConnectionState)

### Purpose

Returns a number representing the security state of the network connection for the current file.

### Format

Get (ConnectionState)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 12.0

### Description

Secure connections are encrypted using SSL. If security is important in your environment, your server administrator needs to install a custom SSL certificate.

Returns:

- **0** for no network connection for the current file.
- **1** for a connection that is not encrypted (FileMaker Server with SSL disabled, or to a FileMaker Pro host).
- **2** for a connection that is encrypted but for which the SSL certificate for FileMaker Server cannot be verified. You may be connected to a server pretending to be your actual destination, which could put your confidential information at risk.
- **3** for a connection that is encrypted with a verified SSL certificate.

### Notes

- Get(ConnectionState) is intended for FileMaker Pro or FileMaker Go client connections. If a script is run on FileMaker Server (via FileMaker WebDirect, Custom Web Publishing, FileMaker Server scheduled scripts, or the Perform Script On Server script step), there may be no network connection to verify. In this case, this function returns only 1 or 2.

For FileMaker WebDirect, this function does not reflect the state of the connection between the web browser and FileMaker Server. Instead, the web browser verifies and displays the state of its connection to FileMaker Server.

### Examples

Returns **3** when the connection is encrypted with a verified SSL certificate.

## Get(CurrentDate)

### Purpose

Returns the current date according to the system calendar.

### Format

Get (CurrentDate)

### Parameters

None

### Data type returned

date

### Originated in

FileMaker Pro 6.0 or earlier

### Description

The format of the result of this function varies based on the date format that was in use when the [database file](#) was created. In the United States, dates are generally in the format MM/DD/YYYY. You can change the date and time formats in your operating system.

If the result is displayed in a [field](#), it is formatted according to the date format of the field in the current [layout](#).

---

**Important** To avoid errors when using dates, always use four-digit years. For more information about how FileMaker Pro handles two-digit dates, see [Conversion of dates with two-digit years](#).

---

### Examples

Returns **2/2/2014** when the system date is set to February 2, 2014.

Get (CurrentDate) -Date (1;5;2014) returns **4** when the system date is set to January 9, 2014.

For an example script, see [Exit Application script step](#).

## Get(CurrentExtendedPrivileges)

### Purpose

Returns a list of keywords, based on the account used to evaluate this calculation, for the enabled extended privileges.

### Format

Get (CurrentExtendedPrivileges)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

The keywords in the list are separated by carriage returns. [Extended privileges](#) are additional access rights assigned to an account's [privilege set](#). See [About accounts, privilege sets, and extended privileges](#).

Returns an empty list if a user doesn't have extended privileges assigned for the current database file.

### Notes

- If you specify the [context](#) for the current calculation, this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.
- See also [Get\(AccountExtendedPrivileges\) function](#).

### Examples

The account that is evaluating this calculation uses a privilege set that includes the extended privilege of Access Via FileMaker WebDirect (keyword "fmwebdirect"):

`Position(Get (CurrentExtendedPrivileges); "fmwebdirect"; 1; 1)` returns a value greater than 0.

If you are logged in and running a script that is set to run with full access privileges, `Get (AccountExtendedPrivileges)` returns the extended privileges for your account, but `Get (CurrentExtendedPrivileges)` returns the extended privileges for the Admin account.

## Get(CurrentHostTimestamp)

### Purpose

Returns the host's current date and time (to the nearest second) according to the system clock.

### Format

Get (CurrentHostTimestamp)

### Parameters

None

### Data type returned

timestamp

### Originated in

FileMaker Pro 7.0

### Description

The format of the value returned is determined by the [database file's](#) settings. You can use your [client](#) system's settings in the operating system.

### Notes

- The client machine and [host](#) machine may be in different times zones so `Get (CurrentHostTimestamp)` and `Get (CurrentTimestamp)` may return different date/time values. Also, the current date and time are characteristics of the host system, but the format of the date and time is a characteristic of the database file.

---

**Important** For users who are connected over a network, the `Get (CurrentHostTimestamp)` function can affect the performance of the database file. For example, if you use the function in an unstored calculation field, and the field is visible in a list view, each display of the field requires an additional network access. Stored calculation fields are a better use of the function. For example, if you automatically enter a timestamp for each newly created record using a stored calculation field, you minimize network access.

---

### Examples

Returns **1/1/2014 11:30:01 AM** when the system clock shows January 1, 2014 11:30:01 AM on the host machine.



## Get(CurrentPrivilegeSetName)

### Purpose

Returns the name of the privilege set being used to evaluate this calculation in the file.

### Format

Get (CurrentPrivilegeSetName)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

If a user is using the default Admin account and you haven't modified [access privileges](#) for the file, this function returns **[Full Access]**.

### Notes

- If you specify the [context](#) for the current calculation, this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.
- If this function is evaluated in a script that is set to run with full access privileges, Get(CurrentPrivilegeSetName) returns **[Full Access]**.
- See also [Get\(AccountPrivilegeSetName\) function](#).

### Examples

For current user Administrator, Get (CurrentPrivilegeSetName) might return **[Full Access]**.

For a current user in the sales department, Get (CurrentPrivilegeSetName) might return **[Data Entry Only]**.

For a user with Read-Only Access to a database who is running a script that is set to run with full access privileges, Get (AccountPrivilegeSetName) returns **[Read-Only Access]** but Get (CurrentPrivilegeSetName) returns **[Full Access]** (for the current script).

## Get(CurrentTime)

### Purpose

Returns the current time (to the nearest second) according to the system clock.

### Format

Get (CurrentTime)

### Parameters

None

### Data type returned

time

### Originated in

FileMaker Pro 6.0 or earlier

### Description

The format of the value returned is determined by the operating system settings.

### Notes

- If the calculation result for this script step is set to integer format, it will return the total number of seconds elapsed since the start of the current day.
- In client/server and peer-to-peer environments, Get (CurrentTimestamp) evaluates the status of the client machine running the [script](#) (not the host machine).

### Examples

Returns **11:30:00 AM** when the function result is in text format and the system clock shows 11:30:00 AM. For an example script, see Install OnTimer Script script step.

## Get(CurrentTimestamp)

### Purpose

Returns the current date and time (to the nearest second) according to the system clock.

### Format

Get (CurrentTimestamp)

### Parameters

None

### Data type returned

timestamp

### Originated in

FileMaker Pro 7.0

### Description

The format of the value returned is determined by the operating system settings.

**Note** In client/server and peer-to-peer environments, `Get (CurrentTimestamp)` evaluates the status of the client machine running the [script](#) (not the host machine).

### Examples

Returns **1/1/2014 11:30:00 AM** when the system clock shows January 1, 2014 11:30:00.

## Get(CurrentTimeUTCMilliseconds)

### Purpose

Returns the current time in Coordinated Universal Time (UTC) to the nearest millisecond.

### Format

Get (CurrentTimeUTCMilliseconds)

### Parameters

None

### Data type returned

number, time

### Originated in

FileMaker Pro 13.0

### Description

Returns the current time without time zone adjustments in the form of the number of milliseconds since 1/1/0001. UTC time zone adjustments must be applied to return your local time.

**Note** In client/server and peer-to-peer environments, this function evaluates the status of the client machine running the [script](#) (not the host machine).

### Example 1

Calculates the amount of time it takes to find and sort records as specified in the script.

```
Set Variable [ $START ; Get ( CurrentTimeUTCMilliseconds ) ]
Go to Layout ["Invoices"]
Perform Find [Restore]
Sort Records [Restore; With dialog: Off]
Set Variable [ $END ; Get ( CurrentTimeUTCMilliseconds ) ]
Show Custom Dialog ["Script Duration" ; GetAsTime ( $END - $START ) ]
```

### Example 2

Calculates the local time for the specified time zone. When the time in UTC is 08:43:55.304 and TimeAdjustment contains -7:

```
GetAsTime ( ( Get ( CurrentTimeUTCMilliseconds ) + (
Location::TimeAdjustment * 3600000 ) ) / 1000 ) returns 1:43:55.304 PM.
```

## Get(CustomMenuSetName)

### Purpose

Returns the name of the active custom menu set.

### Format

Get (CustomMenuSetName)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 8.0

### Description

If the active menu set isn't a custom menu set, an empty string is returned.

### Notes

- In FileMaker WebDirect, this function is not supported and returns an empty string.

### Examples

Returns **Custom Menu Set #1** when this custom menu set is active.

Returns an empty string when the **[Standard FileMaker Menus]** menu set is active.

For an example script, see Install Menu Set script step.

## Get(DesktopPath)

### Purpose

Returns the path to the desktop folder for the current user.

### Format

Get (DesktopPath)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 8.0

### Description

In Windows, the path format is /Drive:/Users/UserName/Desktop/.

In OS X, the path format is /DriveName/Users/username/Desktop/.

### Notes

- In FileMaker Server scripts, the Desktop folder is not supported, so this function returns unreliable results. Instead, paths must use the FileMaker Server Documents folder or temporary folder. See [Get/DocumentsPath\) function](#) or [Get\(TemporaryPath\) function](#).
- In FileMaker WebDirect, this function is not supported and returns an empty string.

### Examples

Returns **/C:/Users/John Smith/Desktop/** for a user named John Smith in Windows.

Returns **/Macintosh HD/Users/johnsmith/Desktop/** for a user named John Smith in OS X.

Each part of the path may vary from these examples, so don't rely on the path to follow a particular pattern.

For an example script, see Get Directory script step.

## Get(Device)

### Purpose

Returns a number representing the type of computer or mobile device running the FileMaker product.

### Format

Get (Device)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 13.0

### Description

This function returns a number indicating the type of computer or mobile device currently running FileMaker Pro or FileMaker WebDirect, or the type of iOS device currently running FileMaker Go.

Returns:

- **0** for an unknown device
- **1** for a Mac
- **2** for a computer running Windows
- **3** for iPad
- **4** for iPhone or iPod touch
- **5** for Android

### Examples

Returns **2** if FileMaker Pro or FileMaker WebDirect is currently running on a computer with Windows.

Returns **3** if FileMaker Go or FileMaker WebDirect is currently running on iPad.

For an example script, see AVPlayer Play script step.

## Get/DocumentsPath)

### Purpose

Returns the path to the Documents folder for the current user.

### Format

Get (DocumentsPath)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 8.0

### Description

In Windows, the path format is /Drive:/Users/UserName/Documents/.

In OS X, the path format is /DriveName/Users/username/Documents/.

When running on FileMaker Server, Get/DocumentsPath) returns the location of the Documents folder, which is in the same folder as the server's Backups, Databases, and Scripts folders. The Documents folder is used as a shared location that scripts from different sessions or other processes on the machine can use to import or export files. See [FileMaker Server Help](#).

**Note** In FileMaker WebDirect and Custom Web Publishing, this function is not supported and returns an empty string.

### Examples

For FileMaker Pro, for a user named John Smith, returns:

**/C:/Users/John Smith/Documents/** in Windows.

**/Macintosh HD/Users/johnsmith/Documents/** in OS X

For FileMaker Server, returns:

**/C:/Program Files/FileMaker/FileMaker Server/Data/Documents/** in Windows.

**/Macintosh HD/Library/FileMaker Server/Data/Documents/** in OS X

Each part of the path may vary from these examples, so don't rely on the path to follow a particular pattern.

For an example script, see Set Variable script step.



## Get/DocumentsPathListing)

### Purpose

Returns a list of all files and folders in the Documents folder returned by the Get/DocumentsPath) function.

### Format

Get (DocumentsPathListing)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 10.0

### Description

Each pathname in the Documents folder is listed separated by a line break. Files and folders are named according to FileMaker Pro naming conventions.

Use Get/DocumentsPathListing) with the Import Records script step and Export Records script step to determine if a file exists in the Documents folder before using the Open File script step to open the file. Get/DocumentsPathListing) ensures that multiple scripts can safely read from and write to the same FileMaker Pro database.

**Note** In FileMaker WebDirect and Custom Web Publishing, this function is not supported and returns an empty string.

### Examples

For FileMaker Server, returns the following pathnames:

In Windows:

**/C:/Program Files/FileMaker/FileMaker Server/Data/Documents/lastmonthsales.xlsx**  
**/C:/Program Files/FileMaker/FileMaker Server/Data/Documents/forecastsales.xlsx**  
**/C:/Program Files/FileMaker/FileMaker Server/Data/Documents/SAP**  
**/C:/Program Files/FileMaker/FileMaker Server/Data/Documents/SAP/sap001.txt**  
**/C:/Program Files/FileMaker/FileMaker Server/Data/Documents/SAP/sap002.txt**

In OS X:

**/Macintosh HD/Library/FileMaker Server/Data/Documents/lastmonthsales.xlsx**  
**/Macintosh HD/Library/FileMaker Server/Data/Documents/forecastsales.xlsx**  
**/Macintosh HD/Library/FileMaker Server/Data/Documents/SAP**  
**/Macintosh HD/Library/FileMaker Server/Data/Documents/SAP/sap001.txt**  
**/Macintosh HD/Library/FileMaker Server/Data/Documents/SAP/sap002.txt**

Each part of the path may vary from these examples, so don't rely on the path to follow a particular pattern.

For an example script, see Export Records script step.

## Get(EncryptionState)

### Purpose

Returns a value representing the file's current encryption state.

### Format

Get (EncryptionState)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 13.0

### Description

Returns:

- **0** if the database is not encrypted
- **1** and the Shared ID as a return delimited list if the database is encrypted

### Examples

In an encrypted database file with the Shared ID of 31725, Get (EncryptionState) returns:

- 1
- 31725

## Get(ErrorCaptureState)

### Purpose

Returns 1 (true) if the Set Error Capture script step is on; otherwise, returns 0 (false).

### Format

Get (ErrorCaptureState)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Examples

Returns **1** if the Set Error Capture script step is on.

## Get(FileMakerPath)

### Purpose

Returns the path to the folder of the currently running copy of FileMaker Pro.

### Format

Get (FileMakerPath)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 8.0

### Notes

- In FileMaker WebDirect, this function is not supported and returns an empty string.

### Examples

Returns **/C:/Program Files/FileMaker/FileMaker Pro 15/** in Windows.

Returns **/Macintosh HD/Applications/FileMaker Pro 15/** in OS X.

Each part of the path may vary from these examples, so don't rely on the path to follow a particular pattern.

## Get(FileName)

### Purpose

Returns the name of the currently active file, without the filename extension.

### Format

Get (FileName)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

If the current calculation is stored and you specify its [context](#), this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.

**Note** If you use Get(FileName) in a function parameter that requires a filename and the filename contains a period, include the filename extension in the parameter. Otherwise, functions may interpret the period in the filename as the beginning of the filename extension, which can lead to unexpected results.

### Examples

Returns **Contacts** when Contacts is the active file.

## Get(FilePath)

### Purpose

Returns the full path of the location of the currently active file.

### Format

Get (FilePath)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

In Windows, the full path is file:/drive:/folder/filename for local files. For remote files, the full path is file://volume/folder/filename.

In OS X, the full path is file:/volume/folder/filename for local and remote files.

If you specify the [context](#) for the current calculation, this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.

### Examples

Returns **file:/C:/Users/Username/Documents/Clients.fmp12** for a local file in Windows.

Returns **file:/Sharename/Documents/Clients.fmp12** for remote files in Windows.

Returns **file:/Macintosh HD/Users/Username/Documents/Clients.fmp12** for a local file in OS X.

Returns **fmnet:/192.168.10.10/Clients.fmp12** for a FileMaker Pro networked file.

Each part of the path may vary from these examples, so don't rely on the path to follow a particular pattern.

## Get(FileSize)

### Purpose

Returns the size, in bytes, of the currently active file.

### Format

Get (FileSize)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

If the current calculation is stored and you specify its [context](#), this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.

### Examples

Returns **15000** when the current file size is 15000 bytes.

## Get(FoundCount)

### Purpose

Returns the number of records in the current found set.

### Format

Get (FoundCount)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

If there are multiple windows open in the current [database file](#), each window can have its own found count value, but results are returned for only the foreground window.

If you specify the [context](#) for the current calculation, this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.

### Examples

Returns **7** when there are seven records in the current [found set](#). For example scripts, see [If script step](#).



## Get(HighContrastColor)

### Purpose

Returns the current high contrast default color scheme name.

### Format

Get (HighContrastColor)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

This function returns the current high contrast default color scheme name if the high contrast option is enabled in the Windows Ease of Access Center and if high contrast color scheme is active. Returns an empty string if the option is unavailable or inactive, or if this function is used in OS X.

### Notes

- In FileMaker WebDirect, this function is not supported and returns an empty string.

### Examples

Returns **High Contrast White** when the Windows theme is set to High Contrast White.

## Get(HighContrastState)

### Purpose

Returns 1 (true) if high contrast is available and active; otherwise, returns 0 (false).

### Format

Get (HighContrastState)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Returns:

- **0** if the high contrast option (in the Windows Ease of Access Center) is unavailable or inactive, or if the function is used in OS X.
- **1** if the high contrast option is available and active.

### Notes

- In FileMaker WebDirect, this function is not supported and returns an empty string.

## Get(HostApplicationVersion)

### Purpose

Returns the version of FileMaker Pro or FileMaker Server running on the computer hosting the current file.

### Format

Get (HostApplicationVersion)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 9.0

### Description

Displays a value when used with the same or higher version of FileMaker Pro or FileMaker Server software. If the current file is not [shared](#) or [hosted](#), this function returns an empty string. Also returns an empty string when used from the host computer itself.

### Examples

Returns **Pro 15.0.1** when the host computer is running FileMaker Pro 15 version 1.

Returns **ProAdvanced 15.0.1** when the host computer is running FileMaker Pro 15 Advanced version 1.

Returns **Server 15.0.1** when the host computer is running FileMaker Server 15 version 1.

## Get(HostIPAddress)

### Purpose

Returns the IP address of the host computer for the current file.

### Format

Get (HostIPAddress)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 8.0

### Description

If the current file isn't being [hosted](#), an empty string is returned.

**Note** In FileMaker WebDirect, returns the physical [IP address](#) of the host machine.

If IPv4 and IPv6 addresses are available for remotely hosted files, the address is returned in the most common or default format. This might not be the same format that was used when connecting to the host.

If the current calculation is stored and you specify its [context](#), this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.

### Examples

Returns one of the following when the current file is being hosted:

- IPv4: **14.156.13.121**
- IPv6: **[2001:0DB8:85A3:08D3:1319:8A2E:0370:7334]**

**Note** If the host machine has both IPv4 and IPv6 addresses, Get(HostIPAddress) returns only the IP address that the client used to connect to the host.

Returns one of the following when accessing a locally hosted file:

- **127.0.0.1** if connected to 127.0.0.1
- **:::1** if connected to localhost
- **14.156.13.121** if connected to the computer's IP address or evaluated in FileMaker WebDirect.

## Get(HostName)

### Purpose

Returns the registered name of the computer that is hosting the file.

### Format

Get (HostName)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

On the computer that is [hosting](#) the file, the current registered computer name is displayed in Control Panel (Windows) or System Preferences (OS X).

### Notes

- If the current calculation is stored and you specify its [context](#), this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.

### Examples

Returns **Fred Jones** when Fred Jones is the registered name of the host computer in use.

## Get(InstalledFMPlugins)

### Purpose

Returns the name, version number, and enabled state of installed plug-ins.

### Format

Get (InstalledFMPlugins)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 12.0

### Description

This function is useful for determining whether an installed [plug-in](#) is newer or older than a plug-in required by a file. The version number is returned if it is available.

If multiple plug-ins are installed, Get(InstalledFMPlugins) returns values for each plug-in on separate lines, separated by carriage returns.

Get(InstalledFMPlugins) returns plug-in version information only when plug-in developers have entered version information in the resource file (Windows) or the info.plist file (OS X).

OS X: Plug-ins are stored as packages.

The enabled state is returned as follows:

- **Enabled** The plug-in is enabled in the FileMaker Pro preferences and can be loaded.
- **Disabled** The plug-in is disabled in the FileMaker Pro preferences and cannot be loaded.
- **Ignored** The plug-in failed to load, which could be due to software incompatibility.

### Examples

When:

MyPlugin1 is installed and is enabled in the Plug-ins tab in the Preferences dialog box.

MyPlugin2 is installed and is disabled in the Plug-ins tab in the Preferences dialog box.

MyPlugin3 could not be loaded.

Get(InstalledFMPlugins) returns:

**MyPlugin1;1.0;Enabled**

**MyPlugin2;1.1;Disabled**

**MyPlugin3; ;Ignored**

## Get(LastError)

### Purpose

Returns a number for the error that occurred in the execution of the most recently executed script step.

### Format

Get (LastError)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Use this function to detect and control the outcome of errors. See FileMaker Pro error codes.

### Notes

- OS X: In FileMaker Pro, if an error occurs while performing an [AppleScript](#) via the Perform AppleScript (OS X) script step, the AppleScript error code will be returned.
- For ODBC imports and Execute SQL script steps, if an error occurs while performing a [SQL](#) query, returns FileMaker error 1408. For detailed information about the error, use the [Get\(LastExternalErrorDetail\)](#) function. If there is no information about the error, returns FileMaker error 1409.
- For working with ODBC data sources in the relationships graph, returns FileMaker error 1408.
- Some [script triggers](#) allow for the activating command or event to be canceled if the script executed by the script trigger returns a False value. When a command or event is canceled this way, the error code is set to 20.
- The following control script steps do not clear the last error condition reported by FileMaker Pro: If, Else, Else If, End If, Loop, Exit Loop If, End Loop, Exit Script, and Halt Script.
- For actions that fail to connect to a host because the host's SSL certificate cannot be verified, this function returns an error code. For detailed information about the error, use the [Get\(LastExternalErrorDetail\)](#) function.

**Tip** To create a script that responds to errors without displaying alerts, use this function with the Set Error Capture script step with the **On** option.

### Examples

Returns **0** when the most recent script step executed successfully.

Returns **401** when no records are found after the Perform Find script step has been executed.

For an example script, see Set Error Capture script step.

## Get(LastExternalErrorDetail)

### Purpose

Returns text about errors external to FileMaker that are returned by Get(LastError).

### Format

Get (LastExternalErrorDetail)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

- For [ODBC](#) imports and Execute SQL script steps, returns a detailed ODBC error message.
- For working with ODBC data sources in the relationships graph, returns the readable error string that is generated by the ODBC driver.
- For actions that attempt to connect to a host securely, returns an SSL certificate error message.

### Examples

For ODBC imports and Execute SQL script steps, returns **[DataDirect][Macintosh ODBC Driver Manager] Data source name not found and no default driver specified (-1)** when a data source name wasn't found and the driver wasn't specified.



## Get(LastMessageChoice)

### Purpose

Returns a number corresponding to the button clicked in an alert message displayed by the Show Custom Dialog script step.

### Format

Get (LastMessageChoice)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Returns:

- **1** for the first button (by default, labeled OK)
- **2** for the second button (by default, labeled Cancel)
- **3** for the third button

### Example 1

For an example script, see Show Custom Dialog script step.

## Get(LayoutAccess)

### Purpose

Returns a number based on record access privileges available through the current layout.

### Format

Get (LayoutAccess)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

You assign the [access privileges](#) in the Custom Layout Privileges dialog box.

Returns:

- **0** if the custom layout privileges of an account's privilege set allow **no access** to **Records via this layout**
- **1** if the custom layout privileges of an account's privilege set allow **view only** access to **Records via this layout**. If the database is opened with read-only access, FileMaker Pro returns **1** even if the [layout](#) has read-write access privileges
- **2** if the custom layout privileges of an account's privilege set allow **modifiable** access to **Records via this layout**

### Notes

- Get (LayoutAccess) returns information about record access privileges defined for only the current layout. It ignores current [record](#) access privileges for all other layouts. To fully check access through a layout, consider the return values of Get (LayoutAccess) and the [Get\(RecordAccess\) function](#).
- See Editing layouts privileges for more details about limiting access through layouts.

### Examples

Returns **1** when the layout allows view-only access to records.

## Get(LayoutCount)

### Purpose

Returns the total number of layouts in the file.

### Format

Get (LayoutCount)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

Returns **3** when the file has three [layouts](#).

## Get(LayoutName)

### Purpose

Returns the name of the layout currently being displayed.

### Format

Get (LayoutName)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

If there are multiple windows open in the current file, each window can have its own [layout](#) name value, but results are returned for only the foreground window.

### Notes

- You can use the [Get\(LayoutNumber\) function](#) as an alternative to Get(LayoutName) if there are multiple layouts with the same name.

### Examples

Returns **Product List** when the Product List layout is displayed.

Returns **Customer Invoice** when the Customer Invoice layout is displayed.

For an example script, see Perform Script On Server script step.

## Get(LayoutNumber)

### Purpose

Returns the number of the layout (specified in the Manage Layouts dialog box) currently displayed.

### Format

Get (LayoutNumber)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

If there are multiple windows open in the current file, each window can have its own [layout](#) number value, but results are returned for only the foreground window.

### Notes

- You can use Get(LayoutNumber) as an alternative to the [Get\(LayoutName\) function](#) if there are multiple layouts with the same name.

### Examples

Returns **3** when the current layout is third in the list of layouts in Manage Layouts.

## Get(LayoutTableName)

### Purpose

Returns the name of the table displaying records in the current layout.

### Format

Get (LayoutTableName)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

If no windows are open, an empty string is returned.

### Examples

There are two [layouts](#), Teachers Layout and Coaches Layout, with corresponding [tables](#) named Teachers and Coaches in the table Instructors. An [unstored calculation](#) of Get (LayoutTableName) returns **Teachers** when the current layout is Teachers Layout and returns **Coaches** when the current layout is Coaches Layout.

## Get(LayoutViewState)

### Purpose

Returns a number representing the currently active file view.

### Format

Get(LayoutViewState)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Returns:

- **0** (zero) if the database file is in Form View
- **1** if the database file is in List View
- **2** if the database file is in Table View

If there are multiple windows open in the current file, each window can have its own layout view state value, but results are returned for only the foreground window.

## Get(MenuBarState)

### Purpose

Returns a number for the current menu bar state.

### Format

Get (MenuBarState)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 14.0

### Description

In FileMaker Pro, FileMaker Go and FileMaker WebDirect, returns:

- **0** if the menu bar is hidden and unlocked
- **1** if the menu bar is visible and unlocked
- **2** if the menu bar is visible and locked
- **3** if the menu bar is hidden and locked

### Notes

- In FileMaker Pro, the Show/Hide Menubar script step is not supported.

### Example 1

Returns **1** when the menu bar is visible and unlocked.



## Get(ModifiedFields)

### Purpose

Returns a list of fields that have been modified in the current record of the current table.

### Format

Get (ModifiedFields)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 13.0

### Description

Returns a list of fields modified since the last time the record was committed. The list includes stored calculation fields that are dependent on one or more fields that have been modified. Field names are separated by carriage returns.

For a new record, the list is always empty because the record has never been committed.

### Examples

When the Invoices::Customer Name and Invoices::Company fields are modified and the record is open, Get (ModifiedFields) returns:

- Customer Name
- Company

## Get(MultiUserState)

### Purpose

Returns a number representing the level of sharing for the file using FileMaker Network.

### Format

Get (MultiUserState)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Returns:

- **0** when network [sharing](#) is off
- **1** when network sharing is on, you're accessing the database file from the [host](#) computer, and either all users or a specific group of users (based on their privilege set) have network access to the database file
- **2** when network sharing is on, you're accessing the database file from a [client](#) computer, and either all users or a specific group of users (based on their privilege set) have network access to the database file

### Notes

- If the current calculation is stored and you specify its [context](#), this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.

### Examples

Returns **0** when access is denied to other users.

For an example script, see Set Multi-User script step.

## Get(NetworkProtocol)

### Purpose

Returns the name of the network protocol (TCP/IP) that FileMaker is using on this machine.

### Format

Get (NetworkProtocol)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

Returns **TCP/IP**.

## Get(NetworkType)

### Purpose

In FileMaker Go, returns a number representing the type of network being used to access the current file.

### Format

Get (NetworkType)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 13.0

### Description

In FileMaker Go, returns:

- **0** if the current file is a local file on an iOS device
- **1** if the network type is unknown
- **2** for a cellular network
- **3** for a Wi-Fi network

**Note** In other products in the FileMaker product line, this function is not supported and returns an empty string.

### Examples

Returns **3** when the file is being accessed from a Wi-Fi network.

## Get(PageNumber)

### Purpose

Returns a number representing the current page being printed or previewed.

### Format

Get ( PageNumber )

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

If nothing is being printed or previewed, **0** is returned.

### Notes

- In FileMaker WebDirect, this function is not supported and returns an empty string.

### Examples

Returns **4** when page 4 is being printed or previewed.

## Get(PersistentID)

### Purpose

Returns text representing a unique identifier of the computer or device running FileMaker.

### Format

Get (PersistentID)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 12.0

### Description

Returns a unique, unchanging identifier for the computer on which FileMaker Pro is running, the device on which FileMaker Go is running, or the current FileMaker WebDirect session in the form of a 32-digit hexadecimal string. Get(PersistentID) helps you identify devices that access your solution.

### Notes

- If web browser cookies are cleared during a FileMaker WebDirect session, the value returned by Get(PersistentID) changes.
- If a FileMaker Go user restores an iOS backup to a different device, the value returned by Get(PersistentID) changes. If a backup is restored to the same iOS device, the returned value remains the same. For more information, see the FileMaker Knowledge Base.

### Examples

For a FileMaker Pro client or a FileMaker Go client, Get (PersistentID) returns a value such as **78569d0bd40b898a64e7d08ccdea8220**.

## Get(PreferencesPath)

### Purpose

Returns the path to the preferences and default options folder for the current user.

### Format

Get (PreferencesPath)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 8.0

### Description

In Windows, the path format is /Drive:/Users/UserName/AppData/Local/.

In OS X, the path format is /DriveName/Users/UserName/Library/Preferences/.

### Notes

- In FileMaker WebDirect, this function is not supported and returns an empty string.

### Examples

Returns **/C:/Users/John Smith/AppData/Local/** for a user named John Smith in Windows.

Returns **/Macintosh HD/Users/John Smith/Library/Preferences/** for a user named John Smith in OS X.

Each part of the path may vary from these examples, so don't rely on the path to follow a particular pattern.

## Get(PrinterName)

### Purpose

Returns text identifying the default printer name.

### Format

Get(PrinterName)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

In Windows, returns a string with each of these entries separated by a comma:

- the printer name
- the driver name
- the name of the printer port

In OS X, returns a string with these entries separated by the word **on**:

- the queue name of the printer (if provided)
- the [IP address](#) of the printer

If any of this information isn't available, **<Unknown>** is inserted in the result (except for queue name in OS X).

### Notes

- In FileMaker WebDirect, this function is not supported and returns an empty string.

### Examples

Returns **HP LaserJet P4015, winspool, Ne03** in Windows.

Returns **24.109.265.43** in OS X.



## Get(QuickFindText)

### Purpose

Returns the text that is stored in the quick find box.

### Format

Get (QuickFindText)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 11.0

### Description

Use this function to pass the text that's in the quick find box to a script when you override the Perform Quick Find command in a custom menu set.

### Example 1

Performs a quick find using the text entered in the quick find box. If no records are found, returns a custom error message.

```
Perform QuickFind [Get ( QuickFindText )]
If [Get ( LastError ) = 0]
    Go to Layout ["Invoice Details" ( Invoices )]
    Go to Record/Request/Page [First]
Else
    Show Custom Dialog ["There are no invoices containing \" & Get (
QuickFindText ) & "\""]
End If
```

## Get(RecordAccess)

### Purpose

Returns a number based on the current record's access privileges, assigned through the Custom Record Privileges dialog box.

### Format

Get (RecordAccess)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Returns:

- **0** if the custom record privileges of an account's privilege set have neither **View** nor **Edit** privileges set to **yes** for the current record
- **1** if the custom record privileges of an account's privilege set have **View** set to **yes** for the current record, or if **View** is set to **limited** and the calculation defined for limited access returns a value of **true**  
**Note** If both **View** and **Edit** are set to **yes**, Get (RecordAccess) returns **2**
- **2** if the custom record privileges of an account's privilege set have **Edit** set to **yes** for the current record, or if **Edit** is set to **limited** and the calculation defined for limited access returns a value of **true**

### Notes

- Get (RecordAccess) only returns information about the privileges defined for accessing records. It ignores [access privileges](#) assigned through individual [layouts](#). To fully check access to a record, consider the return values of the [Get\(LayoutAccess\) function](#) and Get (RecordAccess).
- If you specify the [context](#) for the current calculation, this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.
- See Editing record access privileges for more details about limiting access to records.

### Examples

Returns **1** when the record access is view-only.

## Get(RecordID)

### Purpose

Returns the unique ID number of the current record.

### Format

Get (RecordID)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

The number returned is a decimal value (an integer) generated by FileMaker Pro when the record is created. It does not change.

### Notes

- If the current calculation is stored and you specify its [context](#), this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.
- Get (RecordID) may not return a consistent value for records in ODBC data sources.

### Examples

Returns a unique ID for the current record.

## Get(RecordModificationCount)

### Purpose

Returns the total number of times changes to the current record have been committed.

### Format

Get (RecordModificationCount)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

To commit changes, you can, for example:

- click out of all [fields](#) (exit the record)
- go to a different record
- enter [Find mode](#)

If multiple windows are open, clicking in another window does not commit the record.

### Notes

- If the current calculation is stored and you specify its [context](#), this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.
- Get (RecordModificationCount) returns **NULL** for ODBC data sources.

### Examples

Returns **0** if the record has not been modified since it was created.

If changes are made to four fields and all four fields are committed together, the result increments by one. If changes are made to four fields and each change is committed separately, the result increments by four.

## Get(RecordNumber)

### Purpose

Returns the number of the current record in the current found set.

### Format

Get (RecordNumber)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

The value returned is determined by the relative place of the record in the [found set](#), and it changes depending on the find criteria and the [sort order](#).

### Notes

- To return a value that uniquely and permanently identifies a record in this [table](#), use `Get (RecordID)`.
- If you specify the [context](#) for the current calculation, this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.

### Examples

Returns **3** when the current record is the third record in a found set.

For an example script, see Perform Script On Server script step.

## Get(RecordOpenCount)

### Purpose

Returns the total number of open records in the current found set that haven't been saved.

### Format

Get (RecordOpenCount )

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 8.0

### Description

This function returns the number of open (uncommitted) records in the current [found set](#), including uncommitted related records, that the current user hasn't saved. If you specify the [context](#) for the current calculation, this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.

### Examples

Returns **4** if there is one record and three related records that haven't been saved.

## Get(RecordOpenState)

### Purpose

Returns a number representing the state of the current record.

### Format

Get (RecordOpenState)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 8.0

### Description

Returns:

- **0** for a closed (committed) record
- **1** for a new record that hasn't been committed
- **2** for a modified record that hasn't been committed

### Notes

- This function returns the state of the current record pertaining to the current user. If you specify the [context](#) for the current calculation, this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.

### Examples

Returns **1** if the current record is a new record that the current user hasn't saved.

For an example script, see Revert Record/Request script step.

## Get(RequestCount)

### Purpose

Returns the total number of find requests defined for the current table.

### Format

Get (RequestCount)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

If multiple windows are open in the current file, the results are returned for only the foreground window of the file in which the calculation is defined.

### Examples

Returns **5** when there are five [find requests](#) defined for the current [table](#).



## Get(RequestOmitState)

### Purpose

Returns 1 (true) if Omit is selected in the current find request; otherwise, returns 0 (false).

### Format

Get (RequestOmitState)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 8.0

### Examples

Returns **1** when **Omit** is selected in the current [find request](#).

## Get(ScreenDepth)

### Purpose

Returns the number of bits needed to represent the color or shade of gray of a pixel on the main screen.

### Format

Get (ScreenDepth)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

A value of 8 represents 256 (equal to  $2^8$ ) colors or shades of gray.

### Notes

- In FileMaker WebDirect, this function is not supported and returns **32**.

### Examples

Returns **32** on a display showing millions ( $2^{32}$ ) of colors.

Returns **16** on a display showing thousands ( $2^{16}$ ) of colors.

Returns **4** on a VGA display.

Returns **1** on a black-and-white display.

## Get(ScreenHeight)

### Purpose

Returns the height, in points, of the screen in which the window of the current file is open.

### Format

Get (ScreenHeight)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

When the window spans more than one screen, this function uses the screen that contains the largest percentage of the window. If there are multiple windows open in the current [database file](#), each window can have its own screen height value, but results are returned for only the foreground window.

### Examples

Returns **480** when the screen resolution is set to 640 x 480.

## Get(ScreenScaleFactor)

### Purpose

Returns the scale factor of the screen in which the current file is open.

### Format

Get (ScreenScaleFactor)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 13.0.2

### Description

In OS X and FileMaker Go, this function returns a scale factor based on the relative pixel density of the screen; for example:

- **1** for a device without a Retina display
- **2** for a Mac with a Retina display or for iPhone 6
- **3** for iPhone 6 Plus

In Windows, this function returns the scale factor specified in Control Panel.

In FileMaker WebDirect, server-side scripts, and Custom Web Publishing, this function returns **1**.

### Example 1

Get (ScreenScaleFactor) returns **2** on iPad with Retina display.

### Example 2

Detects the screen scale factor, then goes to a layout designed for a large scale factor or goes to a standard layout.

```
If [Get ( ScreenScaleFactor ) = 1.5]
    Go to Layout [Invoices Factor 150%]
Else
    Go to Layout [Invoices]
End If
```

## Get(ScreenWidth)

### Purpose

Returns the width, in points, of the screen in which the window of the current file is open.

### Format

Get (ScreenWidth)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

When the window spans more than one screen, this function uses the screen that contains the largest percentage of the window. If there are multiple windows open in the current [database file](#), each window can have its own screen width value, but results are returned for only the foreground window.

### Examples

Returns **640** when the screen resolution is set to 640 x 480.

## Get(ScriptAnimationState)

### Purpose

Returns 1 (true) if animations are enabled for the current script; otherwise, returns 0 (false).

### Format

Get (ScriptAnimationState)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 13.0

### Description

[Animations](#) are off by default while a script is running.

### Examples

Returns **1** when the current script has been set to enable animations using the Set Layout Object Animation script step.

## Get(ScriptName)

### Purpose

Returns the name of the script that is currently running or is paused.

### Format

Get (ScriptName)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

Returns **Print Report** when the Print Report script is running.

Returns **Update Customer** when the Update Customer script is running.

## Get(ScriptParameter)

### Purpose

Returns the script parameter passed into the current script.

### Format

```
Get (ScriptParameter)
```

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

Use this function as part of a calculation evaluated within a [script](#).

Script parameters can be passed into a script via a button, a script trigger, or another script. See the Perform Script script step.

### Example 1

Clears the contents of the field specified in the script parameter and navigates to that field. This script can be attached to multiple buttons, each referencing a different field, to clear the contents of multiple fields.

```
Set Field By Name [Get ( ScriptParameter ) ; ""]
Go to Field [Get ( ScriptParameter )]
```

### Example 2

Uses the Invoices::Customer Name field as the parameter. Invoices for the current customer are returned in a new window with the Invoice Report layout.

#### Main script: Current Customer Invoices

```
Find Matching Records [Replace; Invoices::Customer ID]
#Calls the View Customer Invoices sub-script, defined below
Perform Script ["View Customer Invoices"; Parameter:
Invoices::Customer Name]
```

#### Sub-script: View Customer Invoices

```
New Window [Name: "Customer: " & Get ( ScriptParameter ); Style:
Document]
Go to Layout ["Invoice Report"]
Sort Records [Restore; With dialog: Off]
```



## Get(ScriptResult)

### Purpose

Returns the script result from a performed sub-script.

### Format

`Get (ScriptResult)`

### Parameters

None

### Data type returned

text, number, date, time, timestamp, container

### Originated in

FileMaker Pro 8.0

### Description

Use this function as part of a calculation evaluated within a [script](#). If a [sub-script](#) doesn't return a result, then the contents of the script result will either be empty or the value of any previous sub-script result. To control the value of a sub-script result, use the Exit Script script step in the sub-script and specify a result.

### Examples

In the following example, the Find Customers script returns the results of a find request when it is called from the Do Reports script. Script Find Customers uses the optional script result of the Exit Script script step. Script Do Reports then uses `Get (ScriptResult)` to determine what other script steps should be performed based on the returned result stored in `Get (ScriptResult)`.

#### Find Customers

```
Set Error Capture [On]
Perform Find [Restore]
New Record/Request
Exit Script [Text Result: Get(FoundCount) < 10]
```

#### Do Reports

```
Perform Script [Find Customers]
If [Get(ScriptResult) = 0]
    Show Custom Dialog ["You have created 10 records already."]
End If
```

## Get(SortState)

### Purpose

Returns a value representing the current sort state.

### Format

Get (SortState)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Returns:

- **0** if the records in the active [table](#) are not [sorted](#)
- **1** if the records in the active table are sorted
- **2** if the records in the active table are partially sorted (semi-sorted)

Each window has its own sort state.

### Notes

- When records are imported from another file to a previously found and sorted set, the records in a sorted set may exist in a semi-sorted state. To include the imported records in the sort order, sort the found set after importing.
- If you specify the [context](#) for the current calculation, this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.

### Examples

Returns **1** when the [records](#) in the active table are sorted.

## Get(StatusAreaState)

### Purpose

Returns a number representing the current status toolbar state.

### Format

Get (StatusAreaState)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Returns:

- **0** (zero) if the [status toolbar](#) is hidden
- **1** if the status toolbar is visible
- **2** if the status toolbar is visible and locked
- **3** if the status toolbar is hidden and locked

If there are multiple windows open on the currently active file, then results are returned for only the active window.

### Examples

Returns **1**, when the current status toolbar is visible.

## Get(SystemDrive)

### Purpose

Returns the drive letter (Windows) or volume name (OS X) where the running operating system is located.

### Format

Get(SystemDrive)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 8.0

### Description

In FileMaker WebDirect, this function is not supported and returns an empty string.

### Examples

Returns **/C:/** in Windows when the operating system is on the C: drive.

Returns **/DriveName/** in OS X when the operating system is on a volume named DriveName.

## Get(SystemIPAddress)

### Purpose

Returns a list of the IP addresses of all computers connected to an active NIC (Network Interface Controller) card.

### Format

Get (SystemIPAddress)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

IP addresses are separated by carriage returns.

In FileMaker WebDirect, Get(SystemIPAddress) returns the IP address of the interface used to connect to the host.

### Examples

Suppose a machine has the following active physical interfaces:

- an Ethernet card not connected to a network with an IP address of 10.10.10.10
- a Wi-Fi interface with an IP address of 192.168.1.1
- a VPN connection with an IP address of 172.172.172.172

The function returns:

**192.168.1.1**

**172.172.172.172**

Suppose a machine has the following active physical interfaces:

- an Ethernet card not connected to a network with an IP address of 2001::10
- a Wi-Fi interface with an IP address of 3FFE:FFFF:101::230:6EFF:FE04:D9FF/48
- a VPN connection with an IP address of 2001:0DB8:85A3:08D3:1319:8A2E:0370:7334

The function returns:

**3FFE:FFFF:101::230:6EFF:FE04:D9FF/48**

**2001:0DB8:85A3:08D3:1319:8A2E:0370:7334**

## Get(SystemLanguage)

### Purpose

Returns the language currently set on the current system.

### Format

Get (SystemLanguage)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Get(SystemLanguage) is evaluated on the Formats setting in Control Panel (Windows) or in System Preferences (OS X). The text that is returned is in the English language.

For hosted databases, Get(SystemLanguage) returns the client's current system language.

**Note** In FileMaker WebDirect, Get(SystemLanguage) returns the web browser's current language.

### Examples

Returns **Japanese** when Japanese is the current format for the region. For an example script, see Set Use System Formats script step.

## Get(SystemNICAddress)

### Purpose

Returns the hardware addresses of all NIC (Network Interface Controller) cards connected to the computer.

### Format

Get (SystemNICAddress)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

Values in the list returned by this function are separated by carriage returns. The address consists of 6 bytes displayed in hexadecimal separated by colons.

### Notes

- In FileMaker WebDirect, this function is not supported and returns an empty string.
- In FileMaker Go, this function is not a reliable method for obtaining the address on an iOS device. To identify an iOS device, use the [Get\(PersistentID\) function](#) instead. For more information, see the FileMaker Knowledge Base.

### Examples

Returns **00:07:34:4e:c2:0d**, for example.

## Get(SystemPlatform)

### Purpose

Returns a number representing the current platform.

### Format

Get (SystemPlatform)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Returns:

- **1** if the current platform is Intel-based Macs
- **-2** if the platform is Windows
- **3** if the platform is iOS
- **4** if the platform is FileMaker WebDirect

### Examples

Get (SystemPlatform) returns **-2** when the current platform is a Windows platform.

Abs (Get (SystemPlatform) ) returns **1** when the current platform is OS X.



## Get(SystemVersion)

### Purpose

Returns the version of the current operating system.

### Format

Get (SystemVersion)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Returns:

- **6.1** for Windows 7
- **6.3** for Windows 8.1
- **10.0** for Windows 10
- **10.11** for OS X version 10.11
- **<operating system or device> <web browser> <browser version>** for FileMaker WebDirect

For operating system or device, returns:

- **Win** for a Windows operating system
- **Mac** for an OS X operating system
- **Linux** for a Linux operating system
- **iPad** for iPad
- **iPhone** for iPhone
- **iPod** for iPod touch
- **Android** for an Android device
- **Other** for an unknown operating system or device

For web browser, returns:

- **Safari** for the Safari browser
- **IE** for the Internet Explorer browser
- **Chrome** for the Chrome browser
- **Other** for an unknown browser

For browser version, returns the version of the web browser accessing FileMaker WebDirect.

### Examples

Returns **10.11** when the current operating system is OS X version 10.11.

Returns **iPad Safari 9.0** for iPad using FileMaker WebDirect in Safari version 9.0.

Returns **Win Chrome 45** for a Windows computer using FileMaker WebDirect in Chrome version 45.

## Get(TemporaryPath)

### Purpose

Returns the path to the temporary folder FileMaker Pro uses for the current user, or the path FileMaker Server uses on the system.

### Format

Get (TemporaryPath)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 9.0

### Description

The temporary folder name begins with S, followed by a number representing the session of the database engine during which the operation took place. Because your operating system controls the location of temporary files, the exact path returned may be different from the examples shown. The actual path returned also depends on which product (FileMaker Pro or FileMaker Server) is executing the function.

In FileMaker Pro, the temporary folder and any files placed in it are deleted when FileMaker Pro is terminated. In server-side scripts, each script runs in its own session; once the script is completed, the session terminates and the temporary folder is deleted.

### Notes

- In FileMaker WebDirect, this function is not supported and returns an empty string.

### Examples

Returns:

- Windows: **/C:/Users/username/AppData/Local/Temp/S10/**
- OS X: **/Macintosh HD/var/folders/r8/k3\_nw76d6t95mh6cf3d21fpc0000gn/T/S10/**

Each part of the path may vary from these examples, so don't rely on the path to follow a particular pattern.

For an example script, see Perform Script On Server script step.

## Get(TextRulerVisible)

### Purpose

Returns 1 (true) if the text ruler is displayed; otherwise, returns 0 (false).

### Format

Get (TextRulerVisible)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 8.0

### Notes

- In FileMaker WebDirect, this function is not supported and returns **0**.

### Examples

Returns **1** when the text ruler is visible.

## Get(TotalRecordCount)

### Purpose

Returns the total number of records in the current table.

### Format

Get (TotalRecordCount)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

If the current calculation is stored and you specify its [context](#), this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.

### Examples

Returns **876** when there are 876 records in the current [table](#). For an example script, see Set Next Serial Value script step.

## Get(TouchKeyboardState)

### Purpose

Returns 1 (true) if the touch keyboard is set to display automatically when needed; otherwise, returns 0 (false).

### Format

Get (TouchKeyboardState)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 14.0

### Description

In FileMaker Go and Windows, this function returns:

- **1** if the touch keyboard is enabled
- **0** if the touch keyboard is disabled

In Windows 7, OS X, FileMaker WebDirect, server-side scripts, and Custom Web Publishing, this function is not supported and returns **0**.

### Notes

- In FileMaker Go, the value returned by this function does not indicate the state of the touch keyboard if the active field is a [concealed edit box](#), because FileMaker Go always displays the touch keyboard for a concealed edit box.

### Example 1

Checks the current setting of the touch keyboard and disables the touch keyboard if it is enabled.

```
If [ Get ( TouchKeyboardState ) = 1 ]
    Enable Touch Keyboard [Off]
End If
```

## Get(TriggerCurrentPanel)

### Purpose

Returns the index and object name of the panel to switch from when the OnPanelSwitch script trigger is activated.

### Format

Get (TriggerCurrentPanel)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 12.0

### Description

Use with the [Get\(TriggerTargetPanel\)](#) function. Returns an index value, starting from 1, when running a script triggered by the OnPanelSwitch [script trigger](#), and the object name assigned to the [tab panel](#) or [slide panel](#). Returns **0** if the panel is invalid or if Get(TriggerCurrentPanel) is not used with the OnPanelSwitch script trigger.

You can use the [GetValue](#) function to extract the value you want to use from the index value returned by Get(TriggerCurrentPanel).

### Examples

When the tab panel or slide panel to be switched from is panel number 1, named “Category,” Get (TriggerCurrentPanel) returns:

**1**

**Category**

## Get(TriggerExternalEvent)

### Purpose

In FileMaker Go, returns a number representing the event that activated an OnExternalCommandReceived script trigger.

### Format

Get (TriggerExternalEvent)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 14.0

### Description

In FileMaker Go, returns one of the following numbers:

| Returned value | External event  |
|----------------|---|
| 0              | Unknown   |
| 1              | RemotePlayMedia                                       |
| 2              | RemotePause   |
| 3              | RemoteTogglePlayPause                                 |
| 4              | RemotePlayNext  |
| 5              | RemotePlayPrevious                                    |
| 6              | RemoteSeek (begin or end seeking forward or backward) |
| 7              | RemoteStop  |

### Example 1

Controls which media file to play based on the event that activated the OnExternalCommandReceived script trigger on the lock screen or an external device.

```
If [Get(TriggerExternalEvent) = 4 ]
  Go to Record/Request/Page [Next]
  AVPlayer Play [Object:: "Player"]
Else If [Get(TriggerExternalEvent) = 5 ]
  Go to Record/Request/Page [Previous]
  AVPlayer Play [Object:: "Player"]
End If
```

## Get(TriggerGestureInfo)

### Purpose

Returns details about the gesture that activated an OnGestureTap script trigger.

### Format

Get (TriggerGestureInfo)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 13.0

### Description

FileMaker Pro for Windows and FileMaker Go only: Returns a list containing these items:

- the string **Tap**, indicating the script was started by an OnGestureTap [script trigger](#)
- a value indicating the tap count
- a value indicating how many fingers were used to make the tap
- the x coordinate in the document where the gesture occurred
- the y coordinate in the document where the gesture occurred

The coordinates (0,0) are located at the top-left corner of the topmost layout part.

For multi-finger gestures, the coordinates returned indicate the center point of the gesture.

In List View, the coordinates returned reflect the location of the tap within the entire list. For example, if the layout has a header 50 points in height, a body 200 points in height, and a footer 25 points in height, and there are 1000 records in the list, the entire layout is 275,000 points in height. A tap near the bottom of the list might return the coordinates (21,274065).

This function returns values for the following gestures:

- tap with one, two, or three fingers (iOS)
- double-tap with one finger (iOS)
- tap with two fingers (Windows)

Returns an empty string if this function is executed when no OnGestureTap script trigger has been activated.

**Note** In FileMaker Pro for Windows 7 and FileMaker WebDirect, this function is not supported and returns an empty string.



## Examples

When a script is triggered by a three-finger tap, and the gesture occurred at the coordinates (400,600), this function returns:

**Tap**

**1**

**3**

**400**

**600**

## Get(TriggerKeystroke)

### Purpose

Returns the characters that activated an OnObjectKeystroke or OnLayoutKeystroke script trigger.

### Format

Get (TriggerKeystroke)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 10.0

### Description

Returns a value when running a script triggered by an OnObjectKeystroke or OnLayoutKeystroke [script trigger](#) or running a script called from the triggered script; otherwise, returns an empty string. Multiple characters may be returned when the input comes from an input method editor (IME).

### Notes

- In FileMaker WebDirect, this function is not supported and returns an empty string.

### Examples

The following code displays the text **Processing input...** when a carriage return is entered:

```
If [ Code ( Get(TriggerKeystroke) ) = 13 ]
    Show Custom Dialog ["Processing input..."]
End If
```

## Get(TriggerModifierKeys)

### Purpose

Returns the state of the keyboard modifier keys when an OnObjectKeystroke or OnLayoutKeystroke script trigger was activated.

### Format

Get (TriggerModifierKeys)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 10.0

### Description

Returns a value only when called from a script activated by a [script trigger](#) or from a sub-script called from the triggered script; otherwise, returns an empty string.

- See [Get\(ActiveModifierKeys\)](#) for a description of the values assigned to the keyboard modifier keys.
- See the [Code function](#) for a list of navigational keys and the codes returned to a script activated by this trigger.

Time might elapse between when the keys that activated a script trigger are pressed and the script asks for information on the modifier keys. Use [Get\(TriggerKeystroke\)](#) and [Get\(TriggerModifierKeys\)](#) to capture the keys that were active when the script trigger was activated. Use [Get\(ActiveModifierKeys\)](#) to capture any current keys being pressed.

### Notes

- Windows: Alt and Ctrl key combinations do not activate script triggers.
- OS X: Command key combinations do not activate script triggers.
- In FileMaker WebDirect, this function is not supported and returns an empty string.

## Examples

- The following example will only display a custom dialog box when lowercase "a" is entered:

```
If [Get(TriggerKeystroke)="a" and Get(TriggerModifierKeys)=0]
    Show Custom Dialog ["You entered \"a\"."]
End If
```

- The value **9** is returned when Shift-Option is pressed on a computer running OS X. If the Option and Shift keys are pressed on a Mac when a script is triggered, Get(TriggerModifierKeys) returns **9**, regardless of which modifier keys have been pressed between when the trigger was activated and when the script runs.

## Get(TriggerTargetPanel)

### Purpose

Returns the index and the object name of the panel to switch to when the OnPanelSwitch script trigger is activated.

### Format

Get (TriggerTargetPanel)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 12.0

### Description

Use with the [Get\(TriggerCurrentPanel\)](#) function. Returns an index value, starting from 1, when running a script triggered by the OnPanelSwitch [script trigger](#), and the object name assigned to the [tab panel](#) or [slide panel](#). Returns **0** if the panel is invalid or if Get(TriggerTargetPanel) is not used with the OnPanelSwitch script trigger.

You can use the [GetValue](#) function to extract the value you want to use from the index value returned by Get(TriggerTargetPanel).

### Examples

When the tab panel or slide panel to be switched to is number 2, named “Products,” Get (TriggerTargetPanel) returns:

**2**

**Products**

## Get(UserCount)

### Purpose

Returns the number of clients currently accessing the file.

### Format

Get (UserCount)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Returns:

- **1** if FileMaker network sharing is turned off
- **1 +** the number of clients if FileMaker network sharing is turned on

This function does not count clients accessing the database file via [ODBC](#) or [JDBC](#).

### Notes

- If you specify the [context](#) for the current calculation, this function will be evaluated based on that context; otherwise, it will be evaluated based on the context of the current window.

### Examples

Returns **5** when there are 4 clients accessing the database file.

## Get(UserName)

### Purpose

Returns the name of the FileMaker user, as specified in the General tab of the Preferences dialog box.

### Format

Get (UserName)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

The returned name is user-specified.

---

**Important** For greater security, use the [Get\(AccountName\) function](#) to track and manage user access: a user cannot change the account name used to log in to a database file.

---

In FileMaker WebDirect, this function returns:

- **[WebDirect-<xxxxx>]**, where <xxxxx> is the last five digits of the current persistent ID. See the [Get\(PersistentID\) function](#).
- **[Guest]** when the user logged in with the Guest account.

In server-side scripts, this function returns:

- the schedule name, if the script is run by a FileMaker Server schedule
- **<script name> - <account name> <x>**, where <x> is the number of scripts run since the FileMaker Script Engine was started

### Examples

Returns **Sharon Lloyd** when Sharon Lloyd is the current user. For an example script, see Add Account script step.

## Get(UseSystemFormatsState)

### Purpose

Returns 1 (true) if Use System Formats in the Format menu is on; otherwise, returns 0 (false).

### Format

Get (UseSystemFormatsState)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 8.0

### Notes

- In FileMaker WebDirect, this function is not supported and returns **0**.

### Examples

Returns **1** when **Use System Formats** is on.



## Get(UUID)

### Purpose

Returns text representing a Universally Unique Identifier (UUID).

### Format

Get (UUID)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 12.0

### Description

Returns a unique 16-byte (128-bit) string. For example, you can use this function to generate a unique ID of a record.

For unstored calculations, returns a new string each time Get(UUID) is evaluated.

### Examples

Stored calculation: In a calculation field, specify the calculation `Get (UUID)`. Every new record in the calculation field has a unique ID such as E47E7AE0-5CF0-FF45-B3AD-C12B3E765CD5.

Unstored calculation: In a calculation field, specify the calculation `Get (UUID)`. For **Storage Options**, select **Do not store calculation results**. Every time a record is accessed, a new string is generated.

## Get(WindowContentHeight)

### Purpose

Returns the height, in points, of the window content area.

### Format

Get (WindowContentHeight)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

The content area depends on the current size of the active window but doesn't include the title bar, scroll bars, zoom controls, and page margins. The content area is the space inside these controls. It does not include the [status toolbar](#) if it is currently showing.

### Notes

- In FileMaker WebDirect, the content area includes the menu bar, status toolbar, scroll bars, and footer area.
- In FileMaker Go, the content area includes the layout but doesn't include the status bar, menu bar, or toolbar.

### Example 1

In OS X, returns **400** in when the current window height is 437 and the status toolbar isn't showing.

### Example 2

In OS X, the example below combines `Get (WindowContentHeight)` with `Get (WindowHeight)` to determine the height of the title bar and horizontal scroll bar:

`Get (WindowHeight) - Get (WindowContentHeight)` returns **37** when the window height is 437 and the status toolbar isn't showing.

### Example 3

For iPhone 4-inch devices in portrait orientation, the height is 44 points for the menu bar, 44 points for the toolbar, and 20 points for the status bar. The `Get(WindowContentHeight)` function returns:

**460** when toolbar, menu bar, and status bar are showing

**548** when toolbar and menu bar are hidden but status bar is showing

### Example 4

For iPhone 4-inch devices in landscape orientation, the height is 34 points for the menu bar and 34 points for the toolbar. iPhone devices don't display the status bar in landscape orientation. The `Get(WindowContentHeight)` function returns:

**252** when both toolbar and menu bar are showing

**320** when both toolbar and menu bar are hidden

## Get(WindowContentWidth)

### Purpose

Returns the width, in points, of the window content area.

### Format

Get (WindowContentWidth)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

The content area depends on the current size of the active window but doesn't include the title bar, scroll bars, zoom controls, or page margins. The content area is the space inside these controls.

### Notes

- In FileMaker WebDirect, the content area includes the menu bar, status toolbar, scroll bars, and footer area.
- In FileMaker Go, the content area includes the layout but doesn't include the device status bar, menu bar, or toolbar.

### Examples

Returns **400** in OS X when the current window width is 415.

The example below combines `Get (WindowContentWidth)` with `Get (WindowWidth)` to determine the width of the vertical scroll bar:

`Get (WindowWidth) - Get (WindowContentWidth)` returns **15** in OS X when the window width is 415.

## Get(WindowDesktopHeight)

### Purpose

Returns the height, in points, of the desktop space.

### Format

Get (WindowDesktopHeight)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

In Windows, the desktop space is the area inside the MDI window (sometimes referred to as the client area). This doesn't include any virtual space available through the scrolling of the MDI window.

In OS X, the desktop space is the area on the monitor in which the active window is located excluding menu bars.

In FileMaker WebDirect, the desktop space is the area on the main monitor. In OS X, the main monitor is where the menu bar is located. In Windows, the main monitor is where the taskbar is located.

### Examples

Returns **956** in Windows when there is a single monitor and its MDI is set to 1280 x 1024.

Returns **1178** in OS X when there is a single monitor and its resolution is set to 1900 x 1200.

## Get(WindowDesktopWidth)

### Purpose

Returns the width, in points, of the desktop space.

### Format

Get (WindowDesktopWidth)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

In Windows, the desktop space is the space inside the MDI window (sometimes referred to as the client area).

In OS X, the desktop space is the area on the monitor in which the active window is located excluding menu bars.

In FileMaker WebDirect, the desktop space is the area on the main monitor. In OS X, the main monitor is where the menu bar is located. In Windows, the main monitor is where the taskbar is located.

### Examples

Returns **450** in Windows when there is a single monitor and its MDI is set to 500 x 450.

Returns **600** in OS X when there is a single monitor and its resolution is set to 800 x 600.

## Get(WindowHeight)

### Purpose

Returns the height, in points, of the window on which the script is acting.

### Format

Get (WindowHeight)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

The height of the window is calculated from the top to bottom outer edges of the window. This position doesn't include shadows or other effects applied to windows.

In FileMaker WebDirect, the window height does not include menus or toolbars that are part of the web browser.

### Notes

- The window on which the script is acting may not necessarily be the foreground window.

### Examples

Get (WindowHeight) returns **300** when the current window's height is 300 points.

## Get(WindowLeft)

### Purpose

Returns the horizontal distance, in points, of the outer edge of the window on which the script is acting, relative to the left-most edge of the screen.

### Format

Get(WindowLeft)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

The origin of the reference coordinate system is at the leftmost corner below the menu bar. A negative value indicates the portion of the left side of the window that is hidden.

### Notes

- The window on which the script is acting may not necessarily be the foreground window.
- In FileMaker WebDirect, this function is not supported and returns an empty string.

### Examples

Returns **52** when the outer edge of the active window is 52 points from the left edge of the screen.

Returns **0** when the active window is 0 points from the left edge of the screen.

For an example script, see New Window script step.



## Get(WindowMode)

### Purpose

Returns a number representing the mode the FileMaker product is in when the function is evaluated.

### Format

Get (WindowMode)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Returns:

- **0** for [Browse mode](#)
- **1** for [Find mode](#)
- **2** for [Preview mode](#)
- **3** if printing is in progress
- **4** (FileMaker Pro Advanced) if evaluating the function from the Data Viewer and the current window is in Layout mode

If a [script](#) using this function runs while the file is in [Layout mode](#), FileMaker Pro switches to Browse mode and returns **0**. If there are multiple windows open in the current [database file](#), each window can have its own window mode value, but results are returned for only the foreground window.

### Examples

Returns **2** if the file is in Preview mode when the function is evaluated.

## Get(WindowName)

### Purpose

Returns the name of the window on which the script is acting.

### Format

Get (WindowName)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

Returns an empty string if there is no window.

### Notes

- The window on which the script is acting may not necessarily be the foreground window.
- You can set the window name with the Set Window Title script step.

### Examples

There are two windows, Teachers and Students, displaying the same [layout](#) that includes an [unstored calculation](#) Calc containing `Get (WindowName)`. **Teachers** is returned when the Teachers window is refreshed, and **Students** is returned when the Students window is refreshed.

## Get(WindowOrientation)

### Purpose

Returns a value representing the orientation of the window on which the script is acting.

### Format

Get(WindowOrientation)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 13.0

### Description

In FileMaker Pro and FileMaker Go only, returns:

- **-2** for landscape left
- **-1** for landscape right
- **0** for square (FileMaker Pro and FileMaker WebDirect only)
- **1** for portrait
- **2** for portrait upside down

**Note** The window on which the script is acting may not necessarily be the foreground window.

### Examples

If the window that the current script is acting on is in portrait orientation, `Get(WindowOrientation)` returns **1**.

You have a calculation field named Orientation that uses `Get(WindowOrientation)` to return a value as listed above. You have another calculation field that references the Orientation field and uses the [If function](#), which returns **Portrait** if the Orientation field returns a value greater than 0 and returns **Landscape** if the Orientation field returns a value less than 0:

```
If(Orientation > 0;"Portrait";"Landscape")
```

## Get(WindowStyle)

### Purpose

Returns the style of the window on which the script is acting.

### Format

Get (WindowStyle)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 12.0

### Description

Returns:

- **0** (zero) if the window is a document window
- **1** if the window is a floating document window
- **2** if the window is a dialog window

**Note** The window on which the script is acting may not necessarily be the foreground window.

### Examples

Returns **0** if the current window is a document window when the function is evaluated.

Returns **1** if the current window is a floating document window when the function is evaluated.

Returns **2** if the current window is a dialog window when the function is evaluated.

## Get(WindowTop)

### Purpose

Returns the vertical distance, in points, of the outer edge of the window on which the script is acting, relative to the bottom edge of the menu bar.

### Format

Get (WindowTop)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

The origin of the reference coordinate system is at the leftmost corner below the menu bar. A negative value indicates the portion of the top part of the window that is hidden behind the menu bar.

### Notes

- The window on which the script is acting may not necessarily be the foreground window.
- In FileMaker WebDirect, this function is not supported and returns an empty string.

### Examples

Returns **52** when the outer edge of the active window is 52 points from the menu bar.

Returns **0** when the outer edge of the active window just touches the menu bar.

For an example script, see New Window script step.

## Get(WindowVisible)

### Purpose

Returns 1 (true) if the window is visible; returns 0 (false) if the window is hidden using Hide Window.

### Format

Get(WindowVisible)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

The window can be located outside of the visible screen space and still return **1**. The current window may not necessarily be the foreground window.

### Examples

Returns **1** when the current window is physically visible.

Returns **0** when the current window has been hidden using the Hide Window command in FileMaker Pro.

For an example script, see Close Window script step.

## Get(WindowWidth)

### Purpose

Returns the width, in points, of the window on which the script is acting.

### Format

Get (WindowWidth)

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

The width of the window is calculated from the leftmost to rightmost outer edge of the window. This position doesn't include shadows or other effects applied to windows.

### Notes

- The window on which the script is acting may not necessarily be the foreground window.

### Examples

Returns **300** when the current window is 300 points wide.

## Get(WindowZoomLevel)

### Purpose

Returns the zoom percentage of the current window.

### Format

Get (WindowZoomLevel)

### Parameters

None

### Data type returned

text

### Originated in

FileMaker Pro 8.0

### Description

In Windows, an asterisk appears next to the zoom percentage when **Enlarge window contents to improve readability** is selected in the General tab of the Preferences dialog box.

### Notes

- In FileMaker WebDirect, this function is not supported and returns **100**.

### Examples

Returns **200** when the current window's zoom percentage is set to 200.

Returns **200\*** in Windows when the current window's zoom percentage is set to 200 and **Enlarge window contents to improve readability** is selected.



## Logical functions

Logical [functions](#) test for a condition to evaluate it as true or false. This is known as a [Boolean](#) value. If the condition is true, FileMaker Pro returns a **1**; if the condition is false, FileMaker Pro returns a **0**. You can use the keywords `True` and `False` with logical functions and operators when a Boolean value is needed. Keyword `True` returns **1** and keyword `False` returns **0**.

Logical functions can also evaluate parameters such as text or arithmetic operations that do not make a true or false statement, or in the case of the `GetField` function, return the contents of another [field](#).

Click a function name for details.

| This function                            | Returns  |
|--|--|
| <a href="#">Case</a>                     | One of several possible results based on a series of tests.  |
| <a href="#">Choose</a>                   | One result value, according to the integer value of a test.  |
| <a href="#">Evaluate</a>                 | Evaluates an <a href="#">expression</a> as a calculation.  |
| <a href="#">EvaluationError</a>          | An error code, if any, from an expression.   |
| <a href="#">ExecuteSQL</a>               | Executes an SQL query statement for the specified table occurrence in a FileMaker Pro file.  |
| <a href="#">GetAsBoolean</a>             | 1 (true) if data converts to a non-zero numeric value or if a container field holds data; otherwise, returns 0 (false).  |
| <a href="#">GetAVPlayerAttribute</a>     | The setting of the specified attribute for the audio, video, or image file in a container field.   |
| <a href="#">GetField</a>                 | Evaluates <code>fieldName</code> and returns the contents of the specified field.  |
| <a href="#">GetFieldName</a>             | The fully qualified name of a field reference.   |
| <a href="#">GetLayoutObjectAttribute</a> | The specified attributes of the layout object given by the <code>objectName</code> currently active in the calculation.  |
| <a href="#">GetNthRecord</a>             | The contents of <code>fieldName</code> from the specified record number.   |
| <a href="#">If</a>                       | Either <code>result1</code> or <code>result2</code> , depending on the value of a test.  |
| <a href="#">IsEmpty</a>                  | 1 (true) if a field is empty or other errors occur; otherwise, returns 0 (false).  |
| <a href="#">IsValid</a>                  | 0 (false) if the data is invalid; otherwise, returns 1 (true).   |
| <a href="#">IsValidExpression</a>        | 1 (true) if an expression syntax is correct; otherwise, returns 0 (false).   |
| <a href="#">Let</a>                      | The result of a variable or variables set to the result of the specified expression(s) for the duration of the calculation, or until the script exits or the file is closed. |
| <a href="#">Lookup</a>                   | The value specified in <code>sourceField</code> , using the <a href="#">relationships</a> in the <a href="#">relationships graph</a> .                                       |
| <a href="#">LookupNext</a>               | The next lower or higher value in <code>sourceField</code> when there isn't a matching related value.  |
| <a href="#">Self</a>                     | The content of the object in which the calculation is defined.   |

## Case

### Purpose

Returns one of several possible results based on a series of tests.

### Format

```
Case (test1;result1{;test2;result2;...;defaultResult})
```

### Parameters

test - any text or numeric [expression](#).

result - result corresponding to the expression.

Parameters in braces { } are optional.

### Data type returned

text, number, date, time, timestamp, container

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Case evaluates each test expression in order, and when a True expression is found, returns the value specified in result for that expression.

You can include a default result at the end of the parameter list. If none of the expressions evaluated return True, the Case function returns the value specified for defaultResult. If no default result is supplied, Case returns an empty result.

### Examples

Case (Score >= 90;"Excellent";Score > 50;"Satisfactory";"Needs Improvement") displays **Excellent** when the score is 90 or above, **Satisfactory** when the score is between 50 and 90, and **Needs Improvement** for any other score.

Case (Shipment Method="Ground";2;Shipment Method="Air";10) returns **2** when the Shipment Method field contains Ground, and returns **10** when the Shipment Method field contains Air.

## Choose

### Purpose

Returns one result value, according to the integer value of a test.

### Format

```
Choose(test;result0{;result1;result2...})
```

### Parameters

`test` - Any integer calculation. The calculation result of `test` must be a number that [indexes](#) into the list that follows. Because the index is a 0-based index, the test result must be 0 to access the first result.

`result` - one or more results.

Parameters in braces { } are optional.

### Data type returned

text, number, date, time, timestamp, container

### Originated in

FileMaker Pro 6.0 or earlier

### Description

FileMaker Pro evaluates `test` to obtain an index number, which is used to choose the corresponding ordinal result.

Because `Choose` is a 0-based list, the first item on the list is indexed 0 and the second item on the list is indexed 1. For example, if `test` evaluates to 2, then `result2` is chosen.

### Examples

```
Choose(Rating;"Not Applicable";"Good";"Fair";"Poor")
```

`Rating` is a number field that is empty or holds a value. If `Rating` is empty or 0, the `Choose` function returns **Not Applicable**. If `Rating` is 1, the result is **Good**. If `Rating` is 2, the result is **Fair**, and if it is 3, the result is **Poor**. If `Rating` contains a value that does not map to one of the result parameters, the `Choose` function returns nothing.

## Evaluate

### Purpose

Evaluates an expression as a calculation.

### Format

`Evaluate (expression { ; [field1;field2;field3;... ] })`

### Parameters

`expression` - any [text expression](#) or [text field](#).

`fields` - a list of fields that this function is dependent on. When these fields are modified, the calculation will update its result.

Parameters in braces { } are optional. Notice that the optional field list is enclosed in brackets [ ].

### Data type returned

text, number, date, time, timestamp, container

### Originated in

FileMaker Pro 7.0

### Description

The optional `fields` parameter is a list of fields this calculation is dependent on. If a necessary field isn't listed, modifying that dependent field won't update the result of the calculation.

### Notes

- If a server-side script contains the Evaluate function, use English function names in Evaluate's calculation. Evaluate does not recognize localized function names when it is evaluated in a server-side script. See [About running scripts on FileMaker Server](#).

### Examples

`Evaluate (TextField)` returns **4** when TextField contains 2 + 2.

`Evaluate ("textfield")` returns **2 + 2** when textfield contains 2 + 2.

`Evaluate (GetField ("textfield"))` returns **4** when textfield contains 2 + 2.

`Evaluate (TextField; [Amount])` returns **.80** when TextField contains .08 \* Amount and the Amount field contains 10.00.

`Evaluate ("Let (TaxRate=.05; "& Tax Rate Calculation &")")` returns **.50** when the field Tax Rate Calculation contains SubTotal \* TaxRate where SubTotal is a numeric field that contains 10.00.

The following example shows how to pass named parameters using the Evaluate, Let, and Get (ScriptParameter) functions, allowing access only to variable "a" (the example returns **6**):

```
ScriptParameter = "a = 5; b = 10"
```

```
Evaluate ("Let ( [ " & Get (ScriptParameter) & " ]; a + 1 )")
```

The following example shows how to pass named parameters, allowing access to both variable "a" and "b". The simplified first parameter makes the second parameter more complex (the example returns **6, 12**):

```
ScriptParameter = "a = 5; b = 10"
```

```
Evaluate("Let ( [" & Get (ScriptParameter) & "]; a + 1 & "\", \" & b + 2 )")
```

The following example shows how to pass named parameters, while keeping the ability to check the syntax of the second parameter of the `Let` function (the example returns **6, 12**):

```
ScriptParameter = "a = 5; b = 10"
```

```
Let( [a = Evaluate("Let( [" & Get (ScriptParameter) & "]; a )"),
      b = Evaluate("Let( [" & Get (ScriptParameter) & "]; b )")]; a + 1
& ", " & b + 2 )
```

**Note** The `Evaluate` function evaluates an expression, including field values to be evaluated as a calculation formula. It also allows you to specify field dependencies so that a calculation using the evaluation function can be triggered due to changes in other fields of the same record. This function evaluates user-defined formulas. For example, you can create a formula in the `Total` field that computes state tax:

```
Evaluate(StateTaxFormula) + ShippingCost
```

where the `StateTaxFormula` field contains:

```
SubTotal * 1.0875
```

and the `SubTotal` field contains the subtotal before tax and shipping.

The `Evaluate` function has an optional second parameter, which is a field the calculation is dependent on. When the dependent field contents change, FileMaker Pro re-evaluates the calculation. In the following example, the `Total` calculation will be re-evaluated when `SubTotal` changes:

```
Evaluate(StateTaxFormula; SubTotal) + ShippingCost
```

The dependent parameter can also be useful in other cases. For example,

```
Evaluate("Get (CurrentTimeStamp)"; [FieldB; FieldC])
```

will store a timestamp in the calculation field whenever `FieldB` or `FieldC` changes.

## EvaluationError

### Purpose

Returns an error code, if any, from an expression.

### Format

EvaluationError (expression)

### Parameters

expression - any calculation [expression](#)

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

There are two types of errors: syntax and runtime. A syntax error indicates an invalid calculation. A runtime error, such as Field missing or Record missing, occurs when the calculation currently being run is valid but cannot properly execute. See FileMaker Pro error codes for a list of error codes and messages.

**Note** The EvaluationError function must enclose the Evaluate function to return any syntax errors.

### Examples

EvaluationError (calculationField) returns **102** (Field Missing) when calculationField contains total + 1 and the field total has been deleted or renamed.

EvaluationError (Evaluate (calculationField)) returns **1207** (Unbalanced Parenthesis) when calculationField contains abs (-1 with no closing parenthesis).

## ExecuteSQL

### Purpose

Executes an SQL query statement for the specified table occurrence in a FileMaker Pro file.

### Format

```
ExecuteSQL(sqlQuery; fieldSeparator; rowSeparator {;arguments...})
```

### Parameters

`sqlQuery` - an SQL SELECT statement. The statement can include a Union clause that combines the results of two queries. The statement can contain programmatically generated SQL (dynamic parameters) that indicate where optional arguments are to be used in the query. Use the question mark character (?) to specify a dynamic parameter.

`fieldSeparator` - the character string used as a separator between fields in the result. If an empty string is specified, the separator is a comma. The field separator is not displayed after the last field in the result.

`rowSeparator` - the character string used as a separator between records in the result. If an empty string is specified, the separator is a carriage return. The row separator is not displayed after the last row in the result.

`arguments`- one or more [expressions](#) that are evaluated and used as values for the dynamic parameters in the query statement.

### Data type returned

text

### Originated in

FileMaker Pro 12.0

### Description

ExecuteSQL enables you to execute SQL SELECT statements containing dynamic parameters to safely query FileMaker Pro databases in order to avoid security vulnerabilities through injection attacks.

ExecuteSQL does not recognize relationships created in FileMaker Pro, which gives you flexibility to define relationships in SQL statements and retrieve data from any table, independent of the layout context.

ExecuteSQL cannot be used with SQL statements that modify data or the database schema (such as the Insert Into or Delete Table commands).

If an error occurs during query parsing or execution, FileMaker Pro returns ?.

### Notes

- To apply the correct formatting to dates in an SQL query, use the DATE statement. If you do not use the DATE statement, ExecuteSQL treats dates as literal strings.
- FileMaker Pro returns date, time, and number data in Unicode/SQL format, not in the locale of the operating system or the file.

- ExecuteSQL accepts only the SQL-92 syntax ISO date and time formats with no braces. ExecuteSQL does not accept the ODBC/JDBC format date, time, and timestamp constants in braces.
- FileMaker SQL uses the Unicode binary sort order, which is different from the FileMaker Pro sort order used with language sorting or with the default language-neutral sort order.
- For more details about SELECT statement syntax, supported SQL statements, expressions, and Catalog functions, see [FileMaker ODBC and JDBC Guide](#) and [FileMaker SQL Reference](#).

### Example 1

Suppose a database contains two tables, Employees and Salaries, which are related through the EmpID field.

| Employees |           |                   | Salaries |        |
|-----------|-----------|-------------------|----------|--------|
| EmpID     | Last Name | Department        | EmpID    | Salary |
| 1         | Smith     | Development       | 1        | 98000  |
| 2         | Ogawa     | Development       | 2        | 87000  |
| 3         | Durand    | Quality Assurance | 3        | 86000  |
| 4         | Garcia    | Quality Assurance | 4        | 90000  |
| 5         | Mehmet    | Documentation     | 5        | 89000  |
| 6         | Ferrini   | Marketing         | 6        | 121000 |

**Note** The Employees::EmpID, Salaries::EmpID, and Salaries::Salary fields are number fields.

You can use the ExecuteSQL function to return a field value from a specific record without changing the current record or modifying the found set.

ExecuteSQL ( "SELECT Department FROM Employees WHERE EmpID = 1"; "" ; "" ) returns **Development** regardless of the current record, found set, or layout.

### Example 2

Suppose you want to add a field to the Employees table to display the percentage of an employee's salary relative to the total salaries in a department. Though you could use a calculation in FileMaker Pro to generate this value, you can use the ExecuteSQL function to specify this query using dynamic parameters. The example below uses table aliases for the Employees table (E) and the Salaries table (S) when specifying fields (S.Salary, E.EmpID, and S.EmpID).

Define a calculation field in the Employees table, then use the ExecuteSQL function to specify the following query statement:

```
Round (
  100 * Salaries::Salary / ExecuteSQL (
    "SELECT SUM (S.Salary)
    FROM Employees E
    JOIN Salaries S
    ON E.EmpID = S.EmpID
    WHERE E.Department = ?";
    "" ; "" ; Employees::Department
  ) ;
2 )
```

On each employee record, the calculation field displays the percentage of the employee's salary relative to the sum of the salaries for the employee's department. For example, the record with a Last Name of "Smith" returns **52.97**, and the record with a Last Name of "Mehmet" returns **100**.



## GetAsBoolean

### Purpose

Returns 1 (true) if data converts to a non-zero numeric value or a container field holds data; otherwise, returns 0 (false).

### Format

GetAsBoolean(data)

### Parameters

data - any text, number, date, time, timestamp or container [expression](#), or a [field](#) containing text, a number, date, time, timestamp or container

### Data type returned

number

### Originated in

FileMaker Pro 8.0

### Description

Returns a [Boolean value](#).

### Examples

GetAsBoolean("") returns **0**.

GetAsBoolean("Some text here.") returns **0**.

GetAsBoolean(Container Field) returns **1** when the field named Container Field contains data, or returns **0** when Container Field is empty.

## GetAVPlayerAttribute

### Purpose

Returns the setting of the specified attribute for the audio, video, or image file in a container field.

### Format

GetAVPlayerAttribute (attributeName)

### Parameters

attributeName - the name of a supported attribute (see below).

### Data type returned

text, number

### Originated in

FileMaker Pro 14.0

### Description

This function is used in FileMaker Go. If this function is called when the media file is playing or is paused, it returns a value for the file's current playback state. If the function is called when no media is playing, it returns a value for the state of the media file most recently played. If the function is called when no media file has been played, it returns an empty string or **0**.

### Attributes

| Attribute     | Returns  | Data type returned |
|---------------|--|--------------------|
| all           | All the attributes and their values.   | text               |
| sourceType    | The source type used for audio and video files:<br><b>0</b> (None)<br><b>1</b> (URL)<br><b>2</b> (Field)<br><b>3</b> (Layout object)<br><b>4</b> (Active object) | number             |
| source        | The URL, field name, or layout object name. If sourceType is 4 (active object), then source returns an empty string.   | text               |
| playbackState | A number representing the state of the media playback:<br><b>0</b> (Stopped)<br><b>1</b> (Playing)<br><b>2</b> (Paused)  | number             |

| Attribute          | Returns   | Data type returned |
|--------------------|---|--------------------|
| presentation       | The method used to display the media:<br><b>0</b> (Embedded)<br><b>1</b> (Full Screen)<br><b>2</b> (Full Screen Only)<br><b>3</b> (Audio Only)  | number             |
| position           | The position (in seconds) currently playing in the media.   | number             |
| startOffset        | The starting position of the playback (in seconds).   | number             |
| endOffset          | The end position of the playback (in seconds); returns <b>0</b> if playing to the end of the media.   | number             |
| duration           | The length of time (in seconds) that the audio or video file will play.   | number             |
| triggerEvent       | Indicates why the last OnObjectAVPlayerChange or OnFileAVPlayerChange script triggers were activated:<br><b>0</b> (Internal)<br><b>1</b> (Script)<br><b>2</b> (Remote)  | number             |
| triggerEventDetail | Provides information about the event that activated the last OnObjectAVPlayerChange or OnFileAVPlayerChange script trigger:<br><b>0</b> (Unknown)<br><b>1</b> (RemotePlayMedia)<br><b>2</b> (RemotePause)<br><b>3</b> (RemoteTogglePlayPause)<br><b>4</b> (RemotePlayNext)<br><b>5</b> (RemotePlayPrevious)<br><b>6</b> (RemoteSeek)<br><b>7</b> (RemoteStop)<br><b>8</b> (ScriptPlayMedia)<br><b>9</b> (ScriptChangePresentation)<br><b>10</b> (ScriptTogglePlayPause)<br><b>11</b> (ScriptStop)<br><b>12</b> (ScriptChangeSetting)<br><b>13</b> (InternalTogglePlayPause)<br><b>14</b> (InternalChangePresentation)<br><b>15</b> (InternalSeek)<br><b>16</b> (InternalStop) | number             |
| sequence           | Indicates which media file should be played next:<br><b>0</b> (None)<br><b>-1</b> (Go to previous)<br><b>+1</b> (Go to next)  | number             |
| result             | <b>0</b> if playback ends successfully; returns <b>1</b> if playback ends due to an error.  | number             |
| hideControls       | <b>1</b> (Yes) if the playback controls are hidden; otherwise returns <b>0</b> (No).  | number             |

| Attribute               | Returns   | Data type returned |
|-------------------------|---|--------------------|
| disableInteraction      | <b>1</b> (Yes) if users cannot interact with the playback; otherwise returns <b>0</b> (No).   | number             |
| disableExternalControls | <b>1</b> (Yes) if the iOS playback controls on the lock screen or on the control panel are disabled when the media is playing or is paused; otherwise, returns <b>0</b> (No). | number             |
| pauseInBackground       | <b>1</b> (Yes) if the audio is paused when FileMaker Go is moved to the background; otherwise returns <b>0</b> (No).  | number             |
| imageSourceType         | The source type used for images:<br><b>0</b> (None)<br><b>1</b> (URL)<br><b>2</b> (Field)<br><b>3</b> (Layout object)<br><b>4</b> (Active object)                             | number             |
| imageSource             | The URL, field name, or layout object name for images.  | text               |
| imageDuration           | The length of time (in seconds) that the images should be displayed.  | number             |

### Example 1

Stops playing a media file if it is currently playing.

```
If [GetAVPlayerAttribute("playbackState") = 1
  AVPlayer Set Playback State [Stopped]
End If
```

### Example 2

Checks the duration of a media file and displays a message if it is longer than 30 minutes.

```
If [GetAVPlayerAttribute("duration") > 1800
  Show Custom Dialog ["Exceeds Maximum Duration"; "The current video is
longer than 30 minutes."]
Else
  AVPlayer Play [Field:Library::Video]
End If
```

### Example 3

Plays the media file in full screen for iPhone, and embedded for iPad.

```
If [Get(Device) = 3 //iPad]
  AVPlayer Play [Object: "Container"; Presentation: Start Embedded]
Else If [Get(Device) = 4 //iPhone]
  AVPlayer Play [Object: "Container"; Presentation: Start Full Screen]
End If
```

## GetField

### Purpose

Evaluates `fieldName` and returns the contents of the specified field.

### Format

`GetField(fieldName)`

### Parameters

`fieldName` - any [text expression](#) or text [field](#) that refers to a field's name

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text, number, date, time, timestamp, container

### Originated in

FileMaker Pro 6.0 or earlier

### Description

`GetField` evaluates `fieldName`, matches the result to a field in the table on which the calculation was performed, and returns the contents of that field.

To return values of fields that have the same field name across multiple tables, enclose `fieldName` in double quotation marks, as in `GetField("fieldName")`.

For example, `GetField("Phone")` returns a value from `Customer::Phone` when `GetField` is calculated on the `Customer` table, and returns `Company::Phone` when `GetField` is calculated on the `Company` table.

To return the values of multiple fields, do not enclose `fieldName` in quotation marks, as in `GetField(fieldName)`, and specify a field that contains a value that matches the name of another field in the table. To change the value that `GetField` returns, change the value in the specified field to match the name of a different field in the table.

For example, a table has three fields: `Customer::ContactMethod`, `Customer::Phone`, and `Customer::Email`. When `Customer::ContactMethod` contains "Phone", `GetField(ContactMethod)` returns the contents of `Customer::Phone`. However, when `Customer::ContactMethod` contains "Email", `GetField(ContactMethod)` returns the contents of `Customer::Email`.

## Example 1

If you sort records on a calculation field that's defined by the GetField function, you can dynamically sort records and display subsummary data by changing the value of the field specified by GetField.

Suppose a database has four fields:

- Customer Name, a text field
- City, a text field
- SortSelection, a global field
- SortKey, a calculation field that's defined as `Get ("SortSelection")`

This script goes to the Invoices layout and sorts records on the SortKey field:

```
Go to Layout ["Invoices" (Invoices)]
Sort Records [Restore; With dialog: Off]
#Sort by the SortKey field
Go to Record/Request/Page [First]
```

If SortSelection contains "Customer Name", SortKey returns the values of the Customer Name field, and the found set is sorted by Customer Name. Any subsummary data specified to appear when records are sorted by SortKey is displayed from the Customer Name field.

If SortSelection is changed to contain "City", SortKey returns the values of the City field, the found set is re-sorted by City, and subsummary data is displayed from the City field.

Because the found set is sorted by SortKey, the sort information is automatically updated when the values returned by SortKey change, and you do not have to perform another sort operation.

## GetFieldName

### Purpose

Returns the fully qualified name of a field reference.

### Format

`GetFieldName (fieldName)`

### Parameters

`fieldName` - any [field object](#) or evaluation of a [text expression](#) that refers to a field's name

### Data type returned

text

### Originated in

FileMaker Pro 10.0

### Description

Use this function to get the fully qualified name of `fieldName` (`tableName::fieldName`).

**Note** If you specify the context for the current calculation, this function will be evaluated based on that context. Otherwise, it will be evaluated based on the context of the current window.

### Examples

`GetFieldName (x)` returns the name of a field reference passed into a custom function as parameter `x`.

`GetFieldName (Evaluate (<fieldName>))` returns the name of a field based on the data stored in `<fieldName>`.

`GetFieldName (Evaluate (Get (ActiveFieldName)))` returns the fully qualified name of the field that has the [focus](#) when executed.

## GetLayoutObjectAttribute

### Purpose

Returns the specified attributes of the layout object given by the `objectName` currently active in the calculation.

### Format

```
GetLayoutObjectAttribute (objectName ; attributeName { ; repetitionNumber ; portalRowNumber } )
```

### Parameters

`objectName` - the name of a named layout object on the current layout.

`attributeName` - the name of a supported attribute (see below).

`repetitionNumber` - the repetition number (for repeating fields).

`portalRowNumber` - the number of the row in the portal.

**Note** Parameters in braces { } are optional.

### Data type returned

text

### Originated in

FileMaker Pro 8.5

### Attributes

| Attribute                  | Returns   | Data type returned |
|----------------------------|---|--------------------|
| <code>objectType</code>    | The object's type, in English.  | text               |
| <code>hasFocus</code>      | <b>1</b> if <code>objectName</code> has <b>focus</b> ; otherwise, returns <b>0</b> . Objects that can have the focus are fields, portals, tab panels, slide panels, buttons, popover buttons, popovers, charts, and groups. Also returns <b>1</b> for a portal when a portal row is selected. | number             |
| <code>containsFocus</code> | <b>1</b> if <code>objectName</code> has focus or if it contains an object that has focus; otherwise, returns <b>0</b> . Objects that can contain the focus are fields, portals, tab panels, slide panels, button bars, buttons, popover buttons, popovers, charts, and groups.                | number             |
| <code>isFrontPanel</code>  | <b>1</b> if the target object is the tab or slide panel that is in front.   | number             |
| <code>isActive</code>      | <b>1</b> if the target object is currently active; otherwise, returns <b>0</b> .  | number             |



| Attribute               | Returns   | Data type returned |
|-------------------------|---|--------------------|
| isObjectHidden          | <b>1</b> if <code>objectName</code> is hidden for the current record; otherwise, returns <b>0</b> . Returns <b>1</b> for objects other than popovers that are to the right of the layout boundary.  | number             |
| bounds                  | A list of numeric values, separated by spaces, that describes the placement of the specified object (top-left to bottom-right).   | text               |
| left                    | The left edge coordinate of the specified object.   | number             |
| right                   | The right edge coordinate of the specified object.  | number             |
| top                     | The top edge coordinate of the specified object.  | number             |
| bottom                  | The bottom edge coordinate of the specified object.   | number             |
| width                   | The width (in points) of the specified object.  | number             |
| height                  | The height (in points) of the specified object.   | number             |
| rotation                | The rotation (in degrees) of the specified object.  | number             |
| startPoint,<br>endPoint | A pair of numeric values (horizontal, vertical), separated by spaces, that represent the start point or end point of a line object. Other objects return the top-left point for <code>startPoint</code> and the bottom-right point for <code>endPoint</code> .  | text               |
| source                  | The source description of the specified object as follows.<br><br>web viewers - returns current URL.<br><br>fields - returns the fully qualified field name (table name::field name).<br><br>text objects - returns the text (does not return merge fields).<br><br>portals - returns the related table name.<br><br>graphics - returns image data such as the image filename.<br><br>charts - returns the XML description of a chart object.<br><br>all other objects - returns an empty string. | text               |

| Attribute        | Returns  | Data type returned |
|------------------|--|--------------------|
| content          | <p>The content of the specified object as follows.</p> <p>web viewers - returns the current content (such as HTML code).</p> <p>fields - returns the field data formatted using the specified object's properties.</p> <p>text objects - returns the text (including text from merge fields).</p> <p>graphics - returns image data, such as the name of a file in a <a href="#">container field</a> if the image is stored (in the field or externally), or the reference to the file if the image is unstored.</p> <p>charts - returns the bitmap representation of a chart object.</p> <p>buttons and popover buttons - returns the button's text.</p> <p>all other objects - returns an empty string.</p> | text               |
| enclosingObject  | The <code>objectName</code> of the enclosing layout object; otherwise, returns an empty string. Only groups, tab panels, slide panels, button bars, popover buttons, popovers, and portals can contain other objects.  | text               |
| containedObjects | A list of named objects contained within <code>objectName</code> . Only groups, tab panels, slide panels, button bars, popover buttons, popovers, and portals can contain other objects.   | text               |

## Notes

- In FileMaker WebDirect, when this function acts on a web viewer or a chart, the `content` and `source` attributes return an empty string.
- If objects are set to auto-resize, attributes returned are based on the resized bounds of the object in its current state.
- If objects are located above the status toolbar, negative coordinate values are returned.
- When `repetitionNumber` or `portalRowNumber` is 0, the function behaves as if the parameter was not specified. For `portalRowNumber`, the function returns data from the first portal row. For `repetitionNumber`, the function acts on the first repetition (for returning `content` or `source`) or acts on the entire field as a whole (for returning `bounds`). Both parameters are necessary because you must be able to reference a particular field repetition within a particular portal row.

## Example 1

Stores a web viewer's current URL in the Search::Homepage field.

```
Set Field [Search::Homepage ; GetLayoutObjectAttribute ( "Web Viewer" ; "source" )]
```

## GetNthRecord

### Purpose

Returns the contents of fieldName from the specified record number.

### Format

```
GetNthRecord(fieldName;recordNumber)
```

### Parameters

fieldName - any [related field](#) or [repeating field](#), or an [expression](#) that returns a field or a repeating field

recordNumber - the [record](#) number from which you want data

### Data type returned

text, number, date, time, timestamp, container

### Originated in

FileMaker Pro 8.0

### Description

The result of `GetNthRecord()` will not be updated when the record referred to by `GetNthRecord()` is a record other than the one in which the calculation is currently being evaluated.

`GetNthRecord` of the current table returns the Nth record of the found set according to how the current table is sorted.

`GetNthRecord` of a related table returns the Nth record of the related set (relative to the current record), based on the sort order of the relationship.

### Examples

`GetNthRecord(First Name;2)` returns the contents of the First Name field for record 2 in the current table.

`GetNthRecord(First Name;Get(RecordNumber)+1)` returns the contents of the First Name field for the next record in the current table.

`GetNthRecord(Contacts::First Name;2)` returns the contents of the First Name field for record 2 in the Contacts table.

`GetNthRecord(Contacts::Has Repetitions[2];2)` returns the contents of the second repetition of the Has Repetitions field for record 2 in the Contacts table.

## If

### Purpose

Returns either result1 or result2, depending on the value of a test.

### Format

```
If(test;result1{;result2})
```

### Parameters

test - any numeric value or logical [expression](#)

result1 - expression or [field](#) name

result2 - expression or field name

Parameters in braces { } are optional.

### Data type returned

text, number, date, time, timestamp, container

### Originated in

FileMaker Pro 6.0 or earlier

### Description

If test is True (any non-zero numeric result), FileMaker Pro returns result1. If test is False (0), result2 is returned. If test is False, and no result2 is supplied, If returns an empty result. Test must be an expression that returns either a numeric or [Boolean](#) (True, False) result.

### Notes

- If you have more than two possible results, consider using the [Case](#) function.
- By default, if test refers to a field that doesn't yet contain a value, If returns an empty result. To override this functionality, deselect the **Do not evaluate if all referenced fields are empty** checkbox.

### Examples

If(Country = "USA";"US Tech Support";"International Tech Support") returns **International Tech Support**, if the Country field contains France or Japan. Returns **US Tech Support** if the Country field contains USA.

If(State ="CA";Subtotal \* CA Tax Rate;0) returns the tax if the purchaser is a resident of California; otherwise returns **0**.

## IsEmpty

### Purpose

Returns 1 (true) if a field is empty or if other errors occur; otherwise, returns 0 (false).

### Format

`IsEmpty(field)`

### Parameters

`field` - any [field](#) name, [text expression](#), or numeric expression

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

This function returns 1 (true) if a field is empty, if a related field, a related table, relationship, or a file is missing, or if another error occurs; otherwise, 0 (false).

### Examples

`IsEmpty(OrderNum)` returns **1** if the OrderNum field is empty.

`If(IsEmpty(LastName); "Invalid record"; "")` displays **Invalid Record** if the LastName field is blank, but displays nothing if there is an entry in LastName.

`IsEmpty(Payments::DatePaid)` returns **1** if, for example, the Payments table has been moved or renamed.

`IsEmpty("text")` returns **0**.

## IsValid

### Purpose

Returns 0 (false) if the data is invalid; otherwise, returns 1 (true).

### Format

IsValid(field)

### Parameters

field - any [field](#) name

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Returns 0 (False) if:

- A field contains an invalid value because of a field type mismatch (text in a date field, for example)
- FileMaker Pro cannot locate (temporarily or permanently) the [related table](#) in which the referenced field is defined
- A field has been deleted from a related table, and therefore the references to that field in the source table are invalid

Otherwise, it returns **1** (the data is valid).

**Note** IsValid does not check field validation options defined in the Manage Database dialog box. Define field validation options to make sure that data is correctly entered into fields during data entry and import. See [Defining field validation](#).

### Examples

IsValid(Datefield) returns **0** if there is non-date data in Datefield, for example if text was [imported](#) into it.

IsValid(Amount) returns **0** if there is only text in the number field Amount.

IsValid(table::field) returns **0** if the related table was renamed and the relationship isn't updated with the new filename.

## IsValidExpression

### Purpose

Returns 1 (true) if an expression syntax is correct; otherwise, returns 0 (false).

### Format

`IsValidExpression(expression)`

### Parameters

`expression` - any calculation [expression](#)

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Examples

`IsValidExpression(calculationField)` returns 1 (true) if `calculationField` contains `total + 1`.

`IsValidExpression(calculationField)` returns 0 (false) if `calculationField` contains `abs(-1` with no closing parenthesis.

## Let

### Purpose

Sets a variable or variables to the result of the specified expression(s) for the duration of the calculation, or until the script exits or the file is closed.

### Format

```
Let ( { [ ] var1=expression1 { ; var2=expression2 . . . } ; calculation )
```

### Parameters

*var* - any [variable](#) name, local variable name, or global variable name (see About naming fields for guidelines on naming variables).

*expression* - any calculation [expression](#), [field](#), or [constant](#).

*calculation* - any calculation expression, field, or constant.

Parameters in braces { } are optional.

### Data type returned

text, number, date, time, timestamp, container

### Originated in

FileMaker Pro 7.0

### Description

The `Let` function allows you to assign the results of expressions to variables and return a result that can use those variables. Use `Let` to make complex formulas easier to read by reducing the need for calling functions within functions. `Let` can also make formulas more efficient by storing the result of function calls in variables for re-use within the formula, reducing the number of redundant function calls.

To assign multiple variables in one `Let` function, use a list syntax enclosed in brackets [ ] and separated by semicolons. To make multiple variables easier to read, you can put each variable and the returned calculation on separate lines, though this formatting is not required. For example:

```
Let ( [
    variable = value;
    variable2 = value2
];
calculation )
```

Three types of variables can be declared in a `Let` function:

- `variable` – a variable (with no prefix), which is available only within the function where it is declared
- `$variable` – a local variable (prefixed with \$), which is available only with the script where it is declared
- `$$variable` – a global variable (prefixed with \$\$), which is available anywhere within the file where it is declared



Once defined, local and global variables can be referenced in any calculation within their scope. Local variables defined in a calculation are scoped to the file but are only available when scripts are not running. A local and a global variable (or even two local variables in different scripts) can have the same name but they are treated as different variables and store different values.

You can include an optional repetition number that appears in brackets [ ] immediately after the variable name. For example:

```
Let ( [
    $variable[repetition] = value;
    $$variable2 = value2
];
calculation )
```

The `Let` function sets the variables from left to right. You can use previously defined variables (for example, variables that you defined with the `Set Variable` script step) to define new variable values, and you can nest one `Let` function within another. If you use a previously defined variable within a nested `Let` function, the variable has scope only within the nested function (as if you had defined a completely unique variable). See the `City` example below.

## Examples

`Let ( x = 5; x*x )` returns **25**.

`Let ( [ x = 5; squared = x*x; cubed = squared*x ]; cubed )` returns **125**.

The following example returns **San Francisco - Paris**.

```
Let (
    City = "Paris";
    Let (
        City = "San Francisco";
        City & "-"
    )
    & City )
```

The following example sets a local variable to the current account's privilege set and returns the contents of the variable. If this calculation is used in a script, the local variable would be available for the duration of the script.

```
Let ( $PRIVILEGE_SET = Get(AccountPrivilegeSetName) ; $PRIVILEGE_SET )
returns [Full Access] if it is evaluated by an account with the Full Access privilege set.
```

The following example sets a local variable `counter` at repetition 50 with a value of 120:

```
Let ( $counter[50] = 120; $counter[50]*2 ) returns 240.
```

The following example shows how to pass named parameters using the `Evaluate`, `Let`, and `Get (ScriptParameter)` functions, allowing access only to variable "a" (the example returns **6**):

```
ScriptParameter = "a = 5; b = 10"
Evaluate (
    "Let ( ["
        & Get (ScriptParameter) & "
    ];
    a+1 )"
)
```

The following example shows how to pass named parameters, allowing access to both variable "a" and variable "b". The simplified first parameter makes the second parameter more complex (the example returns **6, 12**):

```
ScriptParameter = "a = 5; b = 10"
Evaluate (
  "Let ( ["
    & Get (ScriptParameter) & "
  ];
  a+1 & "\", \" & b+2 )"
)
```

The following example shows how to pass named parameters while keeping the ability to check the syntax of the second parameter of the `Let` function (the example returns **6, 12**):

```
ScriptParameter = "a = 5; b = 10"
Let ( [
  a = Evaluate (
    "Let ( ["
      & Get (ScriptParameter) & "
    ];
    a )"
  ),
  b = Evaluate (
    "Let ( ["
      & Get (ScriptParameter) & "
    ];
    b )"
  )
];
a+1 & ", " & b+2 )
```

## Lookup

### Purpose

Returns the value specified in sourceField, using the relationships in the relationships graph.

### Format

```
Lookup(sourceField{; failExpression})
```

### Parameters

sourceField - the [field](#) from which the [lookup](#) value is taken.

failExpression - any [expression](#).

Parameters in braces { } are optional.

### Data type returned

text, number, date, time, timestamp, container

### Originated in

FileMaker Pro 7.0

### Description

The result of the optional failExpression is returned if the lookup fails.

For this function to access the contents of the source field, the [tables](#) containing the source field and [calculation field](#) need to be [related](#). Calculations using the Lookup function won't be forced to be [unstored calculations](#).

**Note** Lookup returns ? when the related table is an ODBC data source.

### Examples

There are two tables, People and Company, in a [database file](#) containing the data shown below.

*People table*

| CompanyID | Employee       |
|-----------|----------------|
| 100       | John Smith     |
| 200       | Peter Wong     |
| 300       | Sally Anderson |

*Company table*

| CompanyID | CompanyName | Code  |
|-----------|-------------|-------|
| 100       | Apple       | 91234 |
| 100       | Apple       | 82345 |
| 200       | FileMaker   | 95054 |

The People and Company tables are related using the number field CompanyID. The calculation CompanyName = Lookup(Company::CompanyName; "Not found") defined in the People table will return **Apple** for the first [record](#), **FileMaker** for the second record, and **Not found** for the third record.

## LookupNext

### Purpose

Returns the next lower or higher value in sourceField when there isn't a matching related value.

### Format

LookupNext (sourceField;lower/higherFlag)

### Parameters

sourceField - the [field](#) from which the [lookup](#) value is taken

lower/higherFlag - the keywords `lower` or `higher` denote whether the value from the next lower/higher matching [record](#) must be taken if no related record is found

### Data type returned

text, number, date, time, timestamp, container

### Originated in

FileMaker Pro 7.0

### Description

Returns the value specified in sourceField using the [relationships](#) in the [relationships graph](#). LookupNext is similar to [Lookup](#), except that when the lookup fails, the value from sourceField in the lower or higher matching record will be returned, as specified by lower/higherFlag.

For this function to access the value in sourceField, the [tables](#) containing the source field and [calculation field](#) need to be related. Calculations using the LookupNext function won't be forced to be [unstored calculations](#).

**Note** LookupNext returns ? when the related table is an ODBC data source.

### Examples

In this example, you are shipping several items and the cost of shipping is based on weight ranges. Use the LookupNext function to find which shipping rate applies for an item. Use LookupNext with the `higher` flag instead of `Lookup` because the weight of an item may not exactly match the maximum weight, therefore we want to find the *next* highest value.

There are two tables, Items and Shipping Costs, in a database file containing data as shown below.

*Items table*

| Item  | Weight | Rate Lookup |
|-------|--------|-------------|
| Lamp  | 8      |             |
| Chair | 22     |             |
| Desk  | 60     |             |
| Bed   | 120    |             |

*Shipping Costs table*

| Rate Code | Maximum Weight |
|-----------|----------------|
| A         | 25             |
| B         | 50             |
| C         | 100            |
| D         | 150            |

The two tables are related by `Weight` and `Max Weight`. The calculation field `Rate Lookup` is defined as `Rate Lookup = LookupNext(Shipping Costs::Max Weight; Higher)`.

The `Rate Lookup` calculation field will return **25**, **25**, **100**, and **150** for records 1 to 4. `Rate Lookup` can then be used to get the correct rate code (A, A, C, and D respectively).

| Item  | Weight | Rate Lookup |
|-------|--------|-------------|
| Lamp  | 8      | 25          |
| Chair | 22     | 25          |
| Desk  | 60     | 100         |
| Bed   | 120    | 150         |

## Self

### Purpose

Returns the content of the object in which the calculation is defined.

### Format

Self

### Parameters

None

### Data type returned

text, number, date, time, timestamp

### Originated in

FileMaker Pro 9.0

### Description

The `Self` function provides a way for a calculation to reference the object with which it is associated without having to explicitly reference the object.

Use `Self` to create a single calculation formula that can be applied to different objects. The `Self` function is helpful for conditional formatting calculations and tooltip calculations because it returns the content of the layout object when that object has a value. You can also use the `Self` function in field definition calculations (including auto-enter and validation calculations) to return the value of the corresponding field.

### Examples

This example formula can be used in an object's conditional formatting panel to set text formatting when the number entered in the field is greater than 10.

`self > 10` returns 1 (True) when applied to a layout field object whose value is greater than 10.

Use the following example in a layout object's tooltip calculation to display different tooltip text according to whether or not a value less than zero was entered.

```
if(self < 0; "Value is less than zero"; "Value is zero or greater")
```

## Mobile functions

Mobile [functions](#) are used with FileMaker Go.

Click a function name for details.

| This function                  | Returns  |
|--------------------------------|--|
| <a href="#">Location</a>       | The current latitude and longitude on an iOS device running FileMaker Go and their accuracy.   |
| <a href="#">LocationValues</a> | The current latitude, longitude, and altitude on an iOS device running FileMaker Go; their accuracy; and minutes since the values were returned. |
| <a href="#">RangeBeacons</a>   | Returns a list of iBeacons and their proximity to an iOS device.   |

## Location

### Purpose

Returns the current latitude and longitude on an iOS device running FileMaker Go, and their accuracy.

### Format

`Location (accuracy {; timeout})`

### Parameters

`accuracy` - any numeric expression or field containing a number that represents a distance in meters.

`timeout` - any numeric expression or field containing a number that represents the most time it will take to fetch the location. Measured in seconds, the default value is 60.

Parameters in braces { } are optional.

### Data type returned

Text

### Originated in

FileMaker Pro 12.0

### Description

This function returns the current latitude and longitude on an iOS device running FileMaker Go; returns the horizontal accuracy of the values returned. The location is obtained via GPS, cellular network, or Wi-Fi.

Returns and caches the location of a device in the format `latitude, longitude`. You can use the values that are returned to query map services. `Location` returns the horizontal `accuracy` in meters and fetches the location values until `timeout`. If you cancel the process, FileMaker Go returns the most accurate location in the cache (if any).

If no location is received, FileMaker Go returns an empty string. In FileMaker Pro, `Location` returns an empty string.

**Note** To avoid excessive battery consumption and repeat fetches, specify a smaller number for `timeout`.

### Examples

`Location ( 100; 40 )` takes up to 40 seconds to return the latitude and longitude with an accuracy of 100 meters.

```
+110.230000, -131.340000, +65.000000
```



## LocationValues

### Purpose

Returns the current latitude, longitude, and altitude on an iOS device running FileMaker Go; their accuracy; and minutes since the values were returned.

### Format

LocationValues (accuracy {; timeout})

### Parameters

**accuracy** - any numeric expression or field containing a number that represents a distance in meters.

**timeout** - any numeric expression or field containing a number that represents the most time it will take to fetch the location. Measured in seconds, the default value is 60.

Parameters in braces { } are optional.

### Data type returned

Text

### Originated in

FileMaker Pro 12.0

### Description

This function returns the current latitude, longitude, and altitude on an iOS device running FileMaker Go; returns the horizontal and vertical accuracy of the values returned; and number of minutes since the values were returned. The location is obtained via GPS, cellular network, or Wi-Fi.

Returns and caches the current location of a device in the format:

```
latitude¶longitude¶altitude¶horizontal accuracy (+/- accuracy in meters)¶
vertical accuracy (+/- accuracy in meters)¶age of value in minutes (0.2
would represent 0.2 minutes or 12 seconds ago)
```

You can use the GetValue function to retrieve any of the six carriage return-delimited values above.

LocationValues fetches the location values until the requested accuracy is met or until timeout. If you cancel the process, FileMaker Go returns the most accurate location in the cache (if any). If no location is received, FileMaker Go returns an empty string.

In FileMaker Pro, LocationValues returns an empty string.

**Note** To avoid excessive battery consumption and repeat fetches, specify a larger number for accuracy and a smaller number for timeout.

### Examples

LocationValues returns the following location for a device:

```
37.406489
-121.983428
0.000000
65
-1
0.001236
```

## RangeBeacons

### Purpose

Returns a list of iBeacons and their proximity to an iOS device.

### Format

RangeBeacons (UUID {; timeout; major; minor})

### Parameters

**UUID** - the universally unique identifier for iBeacons to search for. The UUID identifies one or more iBeacons as a specific type or from a specific organization. For example, iBeacons for all branches of a department store could share the same UUID.

**timeout** - the number of seconds to wait before returning a value. If **timeout** is not specified, the function returns a value after five seconds.

**major** - the value identifying a group of iBeacons. For example, all iBeacons in a specific branch of a department store could share the same **major**.

**minor** - the value identifying specific iBeacons within a group of iBeacons. For example, all iBeacons in the toys section of a specific branch of a department store could share the same **minor**.

Parameters in braces { } are optional.

### Data type returned

Text

### Originated in

FileMaker Pro 15.0

### Description

In FileMaker Go, returns:

**UUID** - the universally unique identifier for each identified iBeacon.

**major** - the value identifying a group of iBeacons.

**minor** - the value identifying specific iBeacons within a group of iBeacons.

**proximity** - a number indicating the relative proximity to an iBeacon. Returns:

- **0** (Unknown) The proximity of the iBeacon could not be determined.
- **1** (Immediate) The iBeacon is very close to the iOS device.
- **2** (Near) The iBeacon is relatively close to the iOS device.
- **3** (Far) The iBeacon is far from the iOS device.

**accuracy** - the accuracy of the proximity value, measured in meters from the iBeacon. This value helps you differentiate between iBeacons with the same proximity value. A negative value means the accuracy could not be determined.

**rss** - the received signal strength of the iBeacons, measured in decibels.

## Notes

- If no iBeacons match the criteria, this function returns an empty string.
- If the iOS Location Services setting is turned off, this function returns an empty string.
- If the query is not valid, this function returns a question mark (?).

## Examples

`RangeBeacons ("D9B9EC1F-XXXX-YYYY-80A9-1E39D4CEA95C")` returns information about all nearby iBeacons with the specified UUID:

**D9B9EC1F-XXXX-YYYY-80A9-1E39D4CEA95C, 10, 1, 0, -1.00, 0**

**D9B9EC1F-XXXX-YYYY-80A9-1E39D4CEA95C, 5, 1, 3, 14.68, -79**

**D9B9EC1F-XXXX-YYYY-80A9-1E39D4CEA95C, 5, 2, 3, 18.96, -81**

`RangeBeacons ("D9B9EC1F-XXXX-YYYY-80A9-1E39D4CEA95C" ; 30)` returns the result after 30 seconds.

`RangeBeacons ("D9B9EC1F-XXXX-YYYY-80A9-1E39D4CEA95C" ; 20 ; 10 ; 1)` returns information about iBeacons with the specified UUID with a `major` value of 10 and a `minor` value of 1. It returns the result after 20 seconds.

## Number functions

Number [functions](#) are used to manipulate numeric data.

Click a function name for details.

| This function                | Returns   |
|------------------------------|---|
| <a href="#">Abs</a>          | The absolute value of a number.   |
| <a href="#">Ceiling</a>      | A number rounded up to the next integer.  |
| <a href="#">Combination</a>  | The number of ways to uniquely choose numberOfChoices items from a set of size setSize.   |
| <a href="#">Div</a>          | The next lowest integer value after dividing a number by a divisor.                       |
| <a href="#">Exp</a>          | The value of the constant e raised to the power of a number.                              |
| <a href="#">Factorial</a>    | The factorial of a number stopping at 1 or at the optional numberOfFactors.               |
| <a href="#">Floor</a>        | A number rounded down to the next lower integer.  |
| <a href="#">Int</a>          | The integer part of a number without rounding.  |
| <a href="#">Lg</a>           | The base-2 logarithm of a number.   |
| <a href="#">Ln</a>           | The base-e (natural) logarithm of a number.   |
| <a href="#">Log</a>          | The base-10 (common) logarithm of a number.   |
| <a href="#">Mod</a>          | The remainder after a number is divided by a divisor.                                     |
| <a href="#">Random</a>       | A number between 0 and 1, including 0 but not including 1.                                |
| <a href="#">Round</a>        | A number rounded to the specified precision (number of decimal places).                   |
| <a href="#">SetPrecision</a> | Any math function with a precision of 16 to 400 digits to the right of the decimal point. |
| <a href="#">Sign</a>         | -1 when a number is negative, 0 when zero, and 1 when positive.                           |
| <a href="#">Sqrt</a>         | The square root of a number.  |
| <a href="#">Truncate</a>     | A number truncated to the specified number of decimal places.                             |

## Abs

### Purpose

Returns the absolute value of a number.

### Format

Abs (number)

### Parameters

number - any numeric [expression](#) or [field](#) containing a numeric expression

### Data type returned

number, time

### Originated in

FileMaker Pro 6.0 or earlier

### Description

The absolute value of a number is positive. For example, if a negative number appears in a field, the Abs function removes the minus sign and changes it to a positive value.

### Examples

Abs (-123) returns **123**.

Abs (PriceDifference) returns the positive value of the number in the PriceDifference field.

Abs (TargetDate - ActualDate) returns a positive value for the number of days difference between the values in TargetDate and ActualDate.

## Ceiling

### Purpose

Returns a number rounded up to the next integer.

### Format

`Ceiling(number)`

### Parameters

`number` - any numeric [expression](#) or [field](#) containing a numeric expression

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Examples

`Ceiling(1.25)` returns **2**.

`Ceiling(-1.25)` returns **-1**.

## Combination

### Purpose

Returns the number of ways to uniquely choose numberOfChoices items from a set of size setSize.

### Format

Combination(setSize;numberOfChoices)

### Parameters

setSize - any numeric [expression](#) or [field](#) containing a non-negative numeric expression

numberOfChoices - any numeric expression or field containing a non-negative numeric expression

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

This function is useful in statistics, combinatorics, and polynomial expansions. The values returned by this function are referred to as combination coefficients. They form Pascal's triangle.

$$\text{Combination} = \frac{\text{Factorial}(\text{setSize, numberOfChoices})}{\text{Factorial}(\text{numberOfChoices})}$$

### Examples

Combination(5;2) returns **10** for a set consisting of {a, b, c, d, e} because the unique choices when choosing two at a time are {ab, ac, ad, ae, bc, bd, be, cd, ce, de}.

(13 \* 12 \* Combination(4;2) \* Combination(4;3)) / Combination(52;5) returns **0.00144057...**, which is the probability of being dealt a full-house in 5-card poker (less than a 1% chance).

## Div

### Purpose

Returns the next lowest integer value after dividing a number by a divisor.

### Format

`Div(number;divisor)`

### Parameters

`number` - any numeric [expression](#) or [field](#) containing a numeric expression

`divisor` - any numeric expression or field containing a numeric expression

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

Equivalent to `Floor(number/divisor)`.

### Examples

`Div(2.5;2)` returns **1**.

`Div(-2.5;2)` returns **-2**.



## Exp

### Purpose

Returns the value of the constant e raised to the power of a number.

### Format

Exp (number)

### Parameters

number - any numeric [expression](#) or [field](#) containing a numeric expression

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

This function returns the value of the constant e (the base of the natural logarithm, equal to 2.7182818) raised to the power of a number.

The Exp function is the inverse of the Ln function.

### Examples

Exp (1) returns **2.71828182....**

Exp (Ln (2) ) returns **2**.

Exp (0) returns **1**.

## Factorial

### Purpose

Returns the factorial of a number, stopping at 1 or at the optional numberOfFactors.

### Format

Factorial (number { ; numberOfFactors } )

### Parameters

number - numeric [expression](#) or [field](#) containing a positive integer.

numberOfFactors - any numeric expression or field containing a number that represents how many factors to include in the multiplication.

Parameters in braces { } are optional.

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

This function is useful in statistics and combinatorics.

Where  $n$  = number and  $i$  = numberOfFactors:

**Factorial( $n$ )** =  $n(n-1)(n-2)\dots(1)$

**Factorial( $n$ ;  $i$ )** =  $n(n-1)(n-2)\dots(n-i+1)$

### Examples

Factorial(3) returns **6**, which =  $3 * 2 * 1$ .

Factorial(10;3) returns **720**, which =  $10 * 9 * 8$ .

## Floor

### Purpose

Returns a number rounded down to the next lower integer.

### Format

Floor (number)

### Parameters

number - any numeric [expression](#) or [field](#) containing a numeric expression

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Examples

Floor (1.25) returns **1**.

Floor (-1.25) returns **-2**.

## Int

### Purpose

Drops digits to the right of the decimal point and returns the integer part of a number without rounding.

### Format

Int (number)

### Parameters

number - any numeric [expression](#) or [field](#) containing a numeric expression

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

Int (1.45) returns **1**.

Int (-3.9) returns **-3**.

Int (123.9) returns **123**.

Int (Players/3) returns **4**, if Players contains **13**.

## Lg

### Purpose

Returns the base-2 logarithm of a number.

### Format

Lg (number)

### Parameters

number - any numeric [expression](#) or [field](#) containing a numeric expression

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

Number can be any positive value. Negative values return an error. For 0, the Lg function returns nothing because this value is out of the acceptable range.

$$\text{Lg} = \frac{\text{Ln}(\text{number})}{\text{Ln}(2)}$$

### Examples

Lg (1) = **0**

Lg (2) = **1**

Lg (32) = **5**

## Ln

### Purpose

Returns the base-e (natural) logarithm of a number.

### Format

Ln (number)

### Parameters

number - any numeric [expression](#) or [field](#) containing a numeric expression

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Number can be any positive value. Negative values and 0 return an error. The Exp function is the inverse of the Ln function.

### Examples

Ln (2.7182818) returns **.99999998...**

Ln (Exp (5) ) returns **5**.

## Log

### Purpose

Returns the base-10 (common) logarithm of a number.

### Format

Log (number)

### Parameters

number - any positive numeric [expression](#) or [field](#) containing a numeric expression

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Number can be any positive value. Negative values return an error. For 0, the Log function returns nothing because this value is out of the acceptable range.

$$\text{Log} = \frac{\text{Ln}(\text{number})}{\text{Ln}(10)}$$

### Examples

Log (1) returns **0**.

Log (100) returns **2**.

## Mod

### Purpose

Returns the remainder after a number is divided by a divisor.

### Format

`Mod(number;divisor)`

### Parameters

`number` - any numeric [expression](#) or [field](#) containing a numeric expression

`divisor` - numeric expression or field containing a numeric expression

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Use the Mod function when converting units of measure, such as from minutes to hours, to return the number of remaining units.

### Example 1

Calculates 210 divided by 4, with a remainder of 2.

`Mod ( 210 ; 4 )` returns **2**.

### Example 2

Converts the hours portion of a time from 24-hour notation to 12-hour notation. When the 24hTime field contains 16:

`Mod ( 24hTime ; 12 )` returns **4**.

### Example 3

Converts a number of months into a number of years and remaining months. When the NumberOfMonths field contains 31:

`Int ( NumberOfMonths / 12 ) & " years, " & Mod ( NumberOfMonths ; 12 ) & " months"` returns **2 years, 7 months**.



## Random

### Purpose

Returns a number between 0 and 1, including 0 but not including 1.

### Format

Random

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Returns a pseudo-random number in the range (0,1). FileMaker Pro generates a new random number when you:

- insert the `Random` function into a [formula](#)
- cause a formula containing the `Random` function to be reevaluated (by changing data in any of the fields the formula uses)
- display or access a [calculation field](#) defined to have an [unstored](#) result

### Example 1

`Round ( Random * 10 ; 0 )` returns a random number from 0 to 9.

### Example 2

`Int ( Dice::NumSides * Random ) + 1` returns a randomly chosen side of a single die.

The following script calculates multiple die rolls, adds the rolls to a single variable, then displays the results in a custom dialog.

```

Loop
  Set Variable [$ROLL; Value:$ROLL + ( Int ( Test::NumSides * Random ) + 1 )]
  Set Variable [$COUNTER; Value:$COUNTER + 1]
  Exit Loop If [$COUNTER = Dice::NumDice]
End Loop
Show Custom Dialog [$ROLL]
```

## Round

### Purpose

Returns a number rounded to the specified precision (number of decimal places).

### Format

Round(number;precision)

### Parameters

number - any numeric [expression](#) or [field](#) containing a numeric expression

precision - any numeric expression or field containing a numeric expression

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

If you round a negative number of decimal places, all digits to the right of the decimal point are dropped, and the number is rounded to the nearest tens, hundreds, and so on. The Round function always rounds up at 0.5.

### Examples

Round(123.456;2) returns **123.46**.

Round(14.5;0) returns **15**.

Round(29343.98;-3) returns **29000**.

Round(123.456;-1) returns **120**.

## SetPrecision

### Purpose

Computes any math function with a precision of 16 to 400 digits to the right of the decimal point.

### Format

`SetPrecision(expression;precision)`

### Parameters

`expression` - any numeric [expression](#)

`precision` - any number or numeric expression

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

All functions except trigonometric functions support extended precision. This function doesn't perform a truncation.

### Examples

`SetPrecision(5/9;30)` returns **0.555555555555555555555555555556**.

`SetPrecision(1.321321321321321321321321321;0)` returns **1.3213213213213213**.

`SetPrecision(If(field1>5;Exp(50);Average(5/9;1/7;5/7));25)` returns either

**5184705528587072464087.4533229334853848274691006** if `field1 > 5`, or

**0.4708994708994708994708995** if `field1 <= 5`.

## Sign

### Purpose

Returns -1 when a number is negative, 0 when zero, 1 when positive.

### Format

`Sign(number)`

### Parameters

`number` - any numeric [expression](#) or [field](#) containing a numeric expression

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

`Sign(15.12)` returns **1**.

`Sign(-175)` returns **-1**.

`Sign(BalanceDue)` returns **0**, if BalanceDue is a number field containing **0**.

## Sqrt

### Purpose

Calculates the square root of a number.

### Format

Sqrt (number)

### Parameters

number - any positive number, numeric [expression](#), or [field](#) containing a numeric expression

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Use this function to calculate Sqrt.

**Sqrt** =  $\sqrt{\text{number}}$

### Examples

Sqrt (4) returns **2**.

Sqrt (SquareFeet) returns **6** if the SquareFeet number field contains **36**.

## Truncate

### Purpose

Returns a number truncated to the specified number of decimal places.

### Format

`Truncate (number ; precision)`

### Parameters

`number` - any numeric [expression](#) or [field](#) containing a numeric expression

`precision` - any numeric expression or field containing a numeric expression for the number of decimal places.

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

This function doesn't evaluate digits beyond the specified `precision`. Use the `Round` function to round up or down to the required precision.

### Examples

`Truncate (123.456 ; 2)` returns **123.45**.

`Truncate (-14.6 ; 0)` returns **-14**.

`Truncate (29343.98 ; -3)` returns **29000**.

`Truncate (123.456 ; 4)` returns **123.456**.

`Truncate (29343.98 ; 5)` returns **29343.98**.

## Repeating functions

Repeating [functions](#) perform calculations on [repeating fields](#).

Click a function name for details.

| This function                 | Returns  |
|-------------------------------|--|
| <a href="#">Extend</a>        | Allows a value in a non-repeating field to be used with every repetition in a repeating field. |
| <a href="#">GetRepetition</a> | The contents of the repeating field specified by a number.                                     |
| <a href="#">Last</a>          | The last valid, non-blank value in a field.  |

## Extend

### Purpose

Allows a value in a non-repeating field to be used with every repetition in a repeating field.

### Format

Extend(non-repeatingField)

### Parameters

non-repeatingField - any non-repeating field (a [field](#) defined to contain only one value), or an [expression](#) that returns a reference to one

### Data type returned

text, number, date, time, timestamp, container

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Use the Extend function with calculations involving both repeating and non-repeating fields. Without the Extend function, the value in non-repeatingField is used only with the first repetition in the repeating field.

### Examples

Extend(TaxRate) \* Quantity \* ItemPrice returns **1.197**, **.6606**, and **1.497** when TaxRate contains **.06**; the repeating field Quantity contains **1**, **3**, and **5**; and the repeating field ItemPrice contains **19.95**, **3.67**, and **4.99**.

Item Count \* Extend(if(Company Size > 100; Reduced Price; Price)) returns **\$1250**, **\$500**, and **\$750** when Reduced Price contains \$50; the repeating field Item Count contains 25, 10, and 15; and Company Size is greater than 100. If Company Size is less than 100 and Price contains \$100, this calculation returns **\$2500**, **\$1000**, and **\$1500**.



## GetRepetition

### Purpose

Returns the contents of the repeating field specified by a number.

### Format

```
GetRepetition(repeatingField;number)
```

### Parameters

`repeatingField` - any [repeating field](#), or an [expression](#) that returns a reference to a repeating field

`number` - the field repetition number

### Data type returned

text, number, date, time, timestamp, container

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

`ParcelBids` is a field defined to repeat with ten values and contains the values 2500, 1200, and 1500.

`GetRepetition(ParcelBids;2)` returns **1200**.

`GetRepetition(if(IsEmpty(ParcelBids) ≠ true, ParcelBids, HouseBids);2)` returns **1200**.

`GetRepetition(ParcelBids;5)` returns nothing.

**Note** You can also find the contents of a particular repetition in a repeating field using brackets `[]` as array operators. For example, `ParcelBids[2]` returns **1200**. See [Getting the contents of a repetition in a repeating field](#).

## Last

### Purpose

Returns the last valid, non-blank value in a field.

### Format

Last (field)

### Parameters

field - any [repeating field](#) or [related field](#), or an [expression](#) that returns a reference to a repeating [field](#) or related field

### Data type returned

text, number, date, time, timestamp, container

### Originated in

FileMaker Pro 6.0 or earlier

### Description

If field specifies a repeating field then it returns the last non-blank repetition. If field specifies a related field, then it returns the last non-blank value in the related set.

**Note** The last related value will depend on the way [related records](#) are [sorted](#). If the related records are not sorted, then the Last function returns a value based on the creation order of the [records](#).

### Examples

Last (ParcelBids) returns **1500** if ParcelBids is a number field defined to repeat with ten values and contains the values 2500, 1200, and 1500.

Last (Payments::PaymentDate) returns the payment date in the last matching record in the Payments table.

Last (if (IsEmpty (Company) ; PersonalPhone ; WorkPhone) ) returns the last non-empty phone number from the repeating field PersonalPhone when the Company field is empty. If the Company field is not empty, the function returns the last non-empty phone number from the repeating field WorkPhone.

## Summary functions

Summary [functions](#) produce a summary of all [records](#) in the [found set](#), or [subsummary values](#) for records in different groups. Formulas can contain more than one summary function. Summary functions calculate more slowly than other functions because they generate values for a range of records.

An alternative way to generate similar calculated results is to use aggregate functions to summarize data in [related records](#) (whether or not they appear in a [portal](#)). See [Aggregate functions](#) and information about summarizing data in portals.

Click the function name for details.

| This function              | Returns   |
|----------------------------|---|
| <a href="#">GetSummary</a> | The value of summaryField for the current range of records when the file is sorted by breakField. |

## GetSummary

### Purpose

Returns the value of summaryField for the current range of records when the file is sorted by breakField.

### Format

GetSummary(summaryField;breakField)

### Parameters

summaryField - [field](#) of type [summary](#), or an [expression](#) that returns a reference to one.

breakField - field, or an expression that returns a reference to one. To calculate a [grand summary](#) value, use the same summary field for both the summary field and the [break field](#) parameters.

GetSummary must be set up in the same table as the break field.

### Data type returned

number, date, time, timestamp

### Originated in

FileMaker Pro 6.0 or earlier

### Description

This function produces [subsummary values](#). If the file isn't [sorted](#) by the break field, the result is blank.

When a summary field is also used as the break field, returns the summary field value for the entire [found set](#) of records (a grand summary value).

Use GetSummary to capture summary values when you want to:

- use summary values in a calculation
- display subsummary values in [Browse mode](#) or in a [body part](#)

Calculations using the GetSummary function are [unstored](#).

**Note** You can get similar results using a [self-join](#) relationship and [Aggregate functions](#). For more information, see Summarizing data in portals.

### Examples

GetSummary(Total Sales;Country) returns a summary of all records pertaining to the value in the Country field.

GetSummary(Total Sales, if(Number of Countries > 1, Country, Sales Zone)) returns a summary of Total Sales by Country if Number of Countries is greater than 1. Otherwise, it returns a summary of Total Sales by Sales Zone.

GetSummary(Total Sales;Total Sales) produces a summary of all records (similar to using a summary field, which is a total of total sales).

If(ThisCharge > 3 \* GetSummary(AvgCharge;Customer), "Verify this charge", "") displays **Verify this charge** if the current charge is greater than three times the average charge.

## Text functions

Text [functions](#) can be used to analyze, rearrange, extract, and build text strings. For example, you could use the MiddleWords function to extract specific words from supplied text.

Text functions operate on these parameters:

- fields of type text
- text [constants](#) (in quotation marks)
- [expressions](#) having a text result

Click a function name for details.

| This function                   | Does this   |
|---------------------------------|---|
| <a href="#">Char</a>            | Returns the characters for the Unicode code points in the number.   |
| <a href="#">Code</a>            | Returns the Unicode code points for the characters in the text.   |
| <a href="#">Exact</a>           | Returns 1 (true) if the contents of any two fields match; otherwise, returns 0 (false).   |
| <a href="#">Filter</a>          | Returns from textToFilter only those characters specified in filterText, in the order they were originally entered in textToFilter.             |
| <a href="#">FilterValues</a>    | Returns a text result containing only the values that were provided in filterValues, in the order they were originally entered in textToFilter. |
| <a href="#">Furigana</a>        | Converts Japanese text to Hiragana, Katakana, or Roman text.  |
| <a href="#">GetAsCSS</a>        | Returns text converted to the <a href="#">CSS (Cascading Style Sheets)</a> format.  |
| <a href="#">GetAsDate</a>       | Returns dates in text as <a href="#">field type</a> date.   |
| <a href="#">GetAsNumber</a>     | Returns only the numbers in text, as field type number.   |
| <a href="#">GetAsSVG</a>        | Returns text converted to the SVG (Scalable Vector Graphics) format.  |
| <a href="#">GetAsText</a>       | Returns data as field type text.  |
| <a href="#">GetAsTime</a>       | Returns times or timestamps in text as field type time.   |
| <a href="#">GetAsTimestamp</a>  | Returns text as field type timestamp.   |
| <a href="#">GetAsURLEncoded</a> | Returns text as URL (Uniform Resource Locator) encoding, for use as a URL.  |
| <a href="#">GetValue</a>        | Returns the requested value given by valueNumber from listOfValues.   |
| <a href="#">Hiragana</a>        | Converts Katakana (Hankaku and Zenkaku) in text to Hiragana.  |
| <a href="#">KanaHankaku</a>     | Converts Zenkaku Katakana to Hankaku Katakana.  |
| <a href="#">KanaZenkaku</a>     | Converts Hankaku Katakana to Zenkaku Katakana.  |
| <a href="#">KanjiNumeral</a>    | Converts Arabic numerals to Kanji numerals.   |
| <a href="#">Katakana</a>        | Converts Hiragana to Zenkaku Katakana.  |
| <a href="#">Left</a>            | Returns the specified numberOfCharacters in text, counting from the left.   |
| <a href="#">LeftValues</a>      | Returns values contained in text, according to the specified numberOfValues, counting from the left.  |
| <a href="#">LeftWords</a>       | Returns text containing the specified numberOfWords in text, counting from the left.  |
| <a href="#">Length</a>          | Returns the number of characters in a field, including all spaces, numbers, and special characters.   |
| <a href="#">Lower</a>           | Returns all letters in specified text as lowercase.   |

| <b>This function</b>            | <b>Does this</b>  |
|---------------------------------|---|
| <a href="#">Middle</a>          | Extracts the specified numberOfCharacters from text, starting at the specified character position.  |
| <a href="#">MiddleValues</a>    | Returns text containing the specified numberOfValues in text, starting at startingValue.  |
| <a href="#">MiddleWords</a>     | Returns text containing the specified numberOfWords from specified text, starting at startingWord.  |
| <a href="#">NumToJText</a>      | Converts Arabic numerals to Japanese text.  |
| <a href="#">PatternCount</a>    | Returns the number of occurrences of searchString in text.  |
| <a href="#">Position</a>        | Returns the starting position of the specified occurrence of searchString in text.  |
| <a href="#">Proper</a>          | Returns the first letter of each word in text as uppercase and all other letters as lowercase.  |
| <a href="#">Quote</a>           | Returns the text form of text enclosed in quotation marks.  |
| <a href="#">Replace</a>         | Replaces a string of characters in specified text with replacementText.   |
| <a href="#">Right</a>           | Returns the specified numberOfCharacters in text, counting from the right.  |
| <a href="#">RightValues</a>     | Returns values contained in text, according to the specified numberOfValues, counting from the right.   |
| <a href="#">RightWords</a>      | Returns text containing the specified numberOfWords in text, counting from the right.   |
| <a href="#">RomanHankaku</a>    | Converts from Zenkaku alphanumeric and symbols to Hankaku alphanumeric and symbols.   |
| <a href="#">RomanZenkaku</a>    | Converts from Hankaku alphanumeric and symbols to Zenkaku alphanumeric and symbols.   |
| <a href="#">SerialIncrement</a> | Returns specified text and numbers, with the numbers incremented by the specified amount.   |
| <a href="#">Substitute</a>      | Returns a text string with every occurrence of searchString in specified text replaced by replaceString.  |
| <a href="#">Trim</a>            | Returns text stripped of all leading and trailing spaces.   |
| <a href="#">TrimAll</a>         | Returns a copy of text with specified spaces (such as spaces between text, or non-Roman spaces such as full- and half-width) removed or inserted. |
| <a href="#">Upper</a>           | Returns all letters in specified text as uppercase.   |
| <a href="#">ValueCount</a>      | Returns a count of the total number of values in specified text.  |
| <a href="#">WordCount</a>       | Returns a count of the total number of words in specified text.   |

## Char

### Purpose

Returns the characters for the Unicode code points in the number.

### Format

Char (number)

### Parameters

number - a decimal number representing one or more Unicode code points

### Data type returned

text

### Originated in

FileMaker Pro 10.0

### Description

Each group of five digits in the number is treated as a Unicode code point, and the character for each five-digit group is returned in the text.

If the number is 0, the function returns an empty string.

If the number is between 1 and 99,999, the function returns a single character.

If the number contains more than five digits, the function returns the string of characters represented by those code points.

**Note** Some Unicode characters can be represented by a single code point or multiple code points. For example, the character ä can be represented by the letter **a** plus ¨ (dieresis) or by the single character **ä**. The single code point version of this kind of character is called a precomposed character or a composite character.

### Examples

Char (0) returns **an empty string (“”).**

Char (97) returns **a.**

Char (98) returns **b.**

Char (9800097) returns **ab.**

Char (228) returns **ä.**

Char (77600097) returns **ä.** In this case the number represents two Unicode characters: the letter **a** and the dieresis character. When these two characters appear together in a string they are displayed as a single character.

## Code

### Purpose

Returns the Unicode code points for the characters in the text.

### Format

Code (text)

### Parameters

text - one or more characters

### Data type returned

number

### Originated in

FileMaker Pro 10.0

### Description

Returns the Unicode code points for the characters in `text`. If zero characters are in `text`, returns an empty string.

If one character is in the text, the function returns the code point for that character. If the text contains multiple characters, the Unicode code point for each character is returned as a group of five digits where the code point for the first character is represented by the low five digits, the code point for the second character in the next higher (to the left) five digits, and so forth.

When converting a composite character such as `ä`, the function returns the Unicode code point for the composite character.

The following table shows how navigational characters are reported to a script activated by this trigger:

| Key Pressed    | Is reported as | Notes   |
|----------------|----------------|---|
| backspace      | 8              | Corresponds to Unicode/ASCII code for BS (backspace)  |
| tab            | 9              | Corresponds to Unicode/ASCII code for HT (horizontal tab)   |
| shift-tab      | 9              | The shift can be detected using the value returned from the <a href="#">Get(TriggerModifierKeys) function</a> |
| enter          | 10             | Corresponds to Unicode/ASCII code for LF (linefeed)   |
| return         | 13             | Corresponds to Unicode/ASCII code for CR (carriage return)  |
| escape         | 27             | Corresponds to Unicode/ASCII code for ESC (escape)  |
| left arrow     | 28             | Corresponds to Unicode/ASCII code for FS (file separator)   |
| up arrow       | 29             | Corresponds to Unicode/ASCII code for GS (group separator)  |
| right arrow    | 30             | Corresponds to Unicode/ASCII code for RS (record separator)   |
| down arrow     | 31             | Corresponds to Unicode/ASCII code for US (unit separator)   |
| space          | 32             | Corresponds to Unicode/ASCII code for Space   |
| forward delete | 127            | Corresponds to Unicode/ASCII code for Delete  |



**Note** If there are too many characters to be represented in the FileMaker number field type, the function returns a NaN (Not a Number) value.

## Examples

Code ( "" ) returns an empty string.

Code ( "a" ) returns **97**.

Code ( "b" ) returns **98**.

Code ( "ab" ) returns **9800097**.

Code ( "ä" ) returns **228**.

Code ( "ä" ) (an **a** followed by a dieresis character entered in a separate keystroke) returns **77600097**.

## Exact

### Purpose

Returns 1 (true) if the contents of any two fields match; otherwise, returns 0 (false).

### Format

`Exact (originalText ; comparisonText)`

### Parameters

`originalText` - any [text expression](#), [text field](#), or [container field](#)

`comparisonText` - any text expression, text field, or container field

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

For text to match exactly, the uppercase and lowercase usage must be the same. For container fields, the data must be stored in the same manner (either embedded, or stored by reference).

**Note** When evaluating values, text attributes such as font, styles, and sizes are not considered.

**Tip** If case isn't important, use the `Lower` or `Upper` function on both parameters to process data before checking for an exact match.

### Examples

`Exact ("McDonald" ; "McDonald")` returns 1 (True).

`Exact ("McDonald" ; "MCDONALD")` returns 0 (False).

`Exact (Upper ("McDonald") ; Upper ("MCDONALD"))` returns 1 (True).

`Exact ("John" ; "John ")` returns 0 (False).

`Exact (BillTo ; ShipTo)` returns 1 (True) when the value in `BillTo` is the same as the value in `ShipTo`.

`Exact (Recipient ; Upper (Recipient))` returns 1 (True), when `Recipient` contains JOHNSON.

`Exact (Country ; "Spain")` returns 1 (True) when the `Country` field contains Spain.

## Filter

### Purpose

Returns from textToFilter only those characters specified in filterText, in the order they were originally entered in textToFilter.

### Format

Filter(textToFilter;filterText)

### Parameters

textToFilter - any [text expression](#) or text [field](#)

filterText - the characters to preserve in the specified text

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

If filterText doesn't have any characters, an empty string is returned. The Filter function is case-sensitive.

### Examples

Filter("(408)555-1212";"0123456789") returns **4085551212**.

Filter("AaBb";"AB") returns **AB**.

The following example removes all text from the provided data, then formats the remaining numbers in the preferred phone number formatting:

```
Let(phone = filter(theField;"0123456789"); "(" & left(phone;3) & ")" &
middle(phone;4;3) & "-" & middle(phone;7;4))
```

If theField contains **Work: 408.555.1212** this calculation returns **(408)555-1212**.

## FilterValues

### Purpose

Returns a text result containing only the values that were provided in filterValues, in the order they were originally entered in textToFilter.

### Format

```
FilterValues(textToFilter; filterValues)
```

### Parameters

textToFilter - any [text expression](#) or text [field](#)

filterValues - values that you want to preserve in the specified text

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

If filterValues doesn't have any values, an empty string is returned.

Values are text items separated by carriage returns. A value can be empty, a single character, a word, a sentence, or a paragraph. When you press Enter or Return, you start creating a new value. The last value will be recognized with or without a carriage return.

When the textToFilter or the filterValues parameter is a literal string, you must insert a paragraph character (¶) between each item in the string. To insert a carriage return character, click the ¶ button in the Specify Calculation dialog box.

The FilterValues function is not case-sensitive.

### Examples

FilterValues("Plaid¶Canvas¶Suitcase"; "Plaid¶Canvas") returns

**Plaid**

**Canvas**

FilterValues(ValueListItems("Database"; "Sizes"); "Medium¶Small") returns

**Small**

**Medium**

when a database file named Database has a value list Sizes that contains Small¶Medium¶Large.

## Furigana

### Purpose

Converts Japanese text to Hiragana, Katakana, or Roman text.

### Format

`Furigana (text { ; option } )`

### Parameters

`text` - any [text expression](#) or [text field](#)

`option` - 1 through 5, depending on how you want to convert `text`

### Data type returned

text

### Originated in

FileMaker Pro 14.0

### Description

Japanese text is composed of Kana (Hiragana and Katakana) and Kanji (characters). Kanji characters have multiple readings, and the Unicode ordering of these characters is arbitrary. Thus, to sort Kanji characters in a truly meaningful way, it is necessary to sort by the Kana (phonetic syllabary) reading intended for each character.

The `option` parameter determines whether Furigana converts the specified text to Hiragana, Katakana, or Roman (Romaji) text.

| Option | Converts text to          |
|--------|---------------------------|
| 1      | Hiragana                  |
| 2      | Full-width Katakana       |
| 3      | Full-width Roman (Romaji) |
| 4      | Half-width Katakana       |
| 5      | Half-width Roman (Romaji) |

If `option` is not specified or any value other than 1 through 5, Furigana returns the specified text in Hiragana.

### Example 1

Converts the specified text to Hiragana.

`Furigana ( " 東京都 " )` returns `とうきょうと`.

### Example 2

Converts the specified text to Roman text.

`Furigana ( " 東京都 " ; 5 )` returns `toukyouto`.

## GetAsCSS

### Purpose

Returns text converted to the CSS (Cascading Style Sheets) format.

### Format

GetAsCSS (text)

### Parameters

text - any [text expression](#) or text [field](#)

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

GetAsCSS returns the specified text in a span element with [CSS](#) properties.

**Note** The GetAsCSS function does not return formats that are set in the Conditional Formatting dialog box.

### Examples

GetAsCSS (text) returns the example result shown below when the field text contains the word “Frank” and the word Frank has the following text attributes: Font = Helvetica, Font Size = 12 points, Font Color = red, Font Style = bold.

Example result:

```
<span style = “font-family: 'Helvetica';font-size: 12px;color: #FF0000;font-weight: bold;text-align: left;”>Frank</span>
```

## GetAsDate

### Purpose

Returns dates in text as field type date.

### Format

GetAsDate (text)

### Parameters

text - any [text expression](#) or text [field](#) containing text in the same format as the date on the system where the file was created

### Data type returned

date

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Use with formulas involving dates or date functions; use the GetAsDate or Date function to enter a date [constant](#) into a [formula](#). The format of text date must be the same as the date format on the system where the file was created.

You can also use this function to convert the number of days to a date. If you specify a number as the parameter, it has to be between 1 (for 1/1/0001) and 1460970 (for 12/31/4000).

**Note** If the function returns a number instead of a date, go to the Specify Calculation dialog box and make sure the **Calculation result is** date.

---

**Important** To avoid errors when using dates, always use four-digit years. For more information about how FileMaker Pro handles two-digit dates, see [Conversion of dates with two-digit years](#).

---

### Examples

GetAsDate ("03/03/2014") returns **3/3/2014**. You can perform date operations on this result using the [Date function](#).

GetAsDate (735516) returns **10/10/2014**. The number 735516 specifies the number of days since 1/1/0001.

Use the following formula to determine the number of days elapsed between values in two timestamp fields:

GetAsDate (EndDate) - GetAsDate (StartDate) returns **90** if the value in the field EndDate is 4/1/2010 1:00 AM and the value in the field StartDate is 1/1/2010 11:15 PM.

## GetAsNumber

### Purpose

Returns only the numbers in text, as field type number.

### Format

GetAsNumber (text)

### Parameters

text - any [text expression](#) or text [field](#) containing numbers

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Use with formulas involving numbers or numeric functions. This function drops all non-numeric characters from text. If zero numeric characters are in text, returns an empty string.

You can also use this function to convert a date to the number of days. The returned number is the number of days since 1/1/0001.

### Examples

GetAsNumber ("FY98") returns **98**.

GetAsNumber (" \$1,254.50 ") returns **1254.5**.

GetAsNumber ("2 + 2") returns **22**.

GetAsNumber (SerialNumber) returns **35684**, when the value of SerialNumber is TKV35FRG6HH84.

GetAsNumber (DateOfBirth) returns **735516**, when the DateOfBirth field contains 10/10/2014.

GetAsNumber (Passcode) returns an empty string, when the Passcode field contains QTjPLeRMaCV.



## GetAsSVG

### Purpose

Returns text converted to the SVG (Scalable Vector Graphics) format.

### Format

GetAsSVG (text)

### Parameters

text - any [text expression](#) or text [field](#)

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

SVG is an Internet text format similar to [HTML](#) or CSS. SVG supports more text formats than HTML, so SVG will represent what you have typed more accurately.

**Note** The GetAsSVG function does not return formats that are set in the Conditional Formatting dialog box.

### Examples

GetAsSVG (text) returns the example result (below) when the field text contains the word “Frank” and the word Frank has the following text attributes: Font = Helvetica, Font Size = 12 points, Font Color = red, Font Style = bold.

Example result:

```
<stylelist>
<style#0>"font-family: 'Helvetica';font-size: 12px;color:
#FF0000;font-weight: bold;text-align: left;",
Begin: 1, End: 5</style>
</stylelist>
<data>
<span style="0">Frank</span>
</data>
```

## GetAsText

### Purpose

Returns data as field type text.

### Format

GetAsText (data)

### Parameters

data - any number, date, time or timestamp [expression](#), or a [field](#) containing a number, date, time, timestamp, or container

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Use with formulas involving text or text functions. The data returned can be a field type number, date, time, timestamp, or container.

For a container field, GetAsText returns external path information, text (when the container contains text that does not resolve into a valid path), or a question mark (?) if the container data is embedded in the database. For container data stored externally, data is returned in the format shown in the following example:

GetAsText (Container) returns

**remote:cat.jpg**

**size:320,240**

**JPEG:Images/Animals/cat.jpg**

### Examples

GetAsText (45) returns **45**.

"You are " & GetAsText (DaysDelinquent) & " days late." returns **You are 3 days late**. when the value of DaysDelinquent is 3.

"FY" & GetAsText (FiscalYear) returns **FY98**, if the FiscalYear number field contains 98.

## GetAsTime

### Purpose

Returns times or timestamps in text as field type time.

### Format

GetAsTime (text)

### Parameters

text - any [text expression](#) or text [field](#) containing a time

### Data type returned

time

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Use with formulas involving the Time or Timestamp function; use the GetAsTime or the Time function to enter a time constant into a [formula](#). The format of the supplied time must be the same as the time format on the system where the file was created.

### Examples

GetAsTime ("02:47:35") returns **2:47:35** when you select time as the calculation result. You can perform time calculations on this result.

GetAsTime ("02:47:35") returns **1/1/0001 2:47:35** when you select timestamp as the calculation result.

Abs (GetAsTime ("12:15 pm") - CheckOut) returns **3:00:00** when the CheckOut time field contains 3:15 PM.

## GetAsTimestamp

### Purpose

Returns text as field type timestamp.

### Format

GetAsTimestamp(text)

### Parameters

text - any [text expression](#), or text, number, date, or time [field](#)

### Data type returned

timestamp

### Originated in

FileMaker Pro 7.0

### Description

Use with formulas involving timestamps. Text strings must be in the form of a date followed by a time. A number is considered to be the number of seconds since 1/1/0001. There are 86400 seconds in each day.

### Examples

GetAsTimestamp("4/5/2014 4:05:06") returns **4/5/2014 4:05:06 AM**.

GetAsTimestamp(50000) returns **1/1/0001 1:53:20 PM**.

## GetAsURLEncoded

### Purpose

Returns text as URL (Uniform Resource Locator) encoding, for use as a URL.

### Format

GetAsURLEncoded (text)

### Parameters

text - any [text expression](#) or text [field](#)

### Data type returned

text

### Originated in

FileMaker Pro 8.5

### Description

This function removes all styles from `text`. All characters are first converted to UTF-8 format. Characters that are neither letters nor digits, or digits that are in the upper ASCII range, are converted to %HH format (a percent sign followed by the character's hexadecimal value).

See the following website for more information on URL encoding:

<http://www.w3.org>

### Examples

GetAsURLEncoded ("Hello") returns **Hello**.

GetAsURLEncoded ("San Francisco") returns **San%20Francisco**.

GetAsURLEncoded ("français") returns **fran%c3%a7ais**.

## GetValue

### Purpose

Returns the requested value given by `valueNumber` from `listOfValues`.

### Format

```
GetValue(listOfValues;valueNumber)
```

### Parameters

`listOfValues` - a list of carriage return-delimited values

`valueNumber` - the value to return from the list

### Data type returned

text

### Originated in

FileMaker Pro 8.0

### Description

This function is useful in looping scripts or recursive custom calculations.

Values are text items separated by carriage returns. You can place several values together to create a carriage return-delimited list of values. A value can be empty, a single character, a word, a sentence, or a paragraph. When you press Enter or Return, you start creating a new value. The last value will be recognized with or without a carriage return.

When the `listOfValues` parameter is a literal string, you must insert a literal carriage return character (¶) between each item in the string. To insert a literal carriage return character, click the ¶ button in the Specify Calculation dialog box.

### Examples

```
GetValue("London¶Paris¶Hong Kong";2) returns
```

**Paris**

## Hiragana

### Purpose

Converts Katakana (Hankaku and Zenkaku) in text to Hiragana.

### Format

Hiragana (text)

### Parameters

text - any [text expression](#) or text [field](#)

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

Hiragana (" アイウエオ ") returns あいうえお

## KanaHankaku

### Purpose

Converts Zenkaku Katakana to Hankaku Katakana.

### Format

KanaHankaku (text)

### Parameters

text - any [text expression](#) or text [field](#)

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

KanaHankaku ( “ データベース ” ) returns テーブル



## KanaZenkaku

### Purpose

Converts Hankaku Katakana to Zenkaku Katakana.

### Format

KanaZenkaku (text)

### Parameters

text - any [text expression](#) or text [field](#)

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

KanaZenkaku ( " データベース " ) returns データベース

## KanjiNumeral

### Purpose

Converts Arabic numerals to Kanji numerals.

### Format

KanjiNumeral (text)

### Parameters

text - any [text expression](#) or text [field](#)

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

KanjiNumeral (123) returns 一二三

KanjiNumeral ( " 富士見台 2 の 3 の 2 5 " ) returns 富士見台二の三の二五

## Katakana

### Purpose

Converts Hiragana to Zenkaku Katakana.

### Format

Katakana (text)

### Parameters

text - any [text expression](#) or text [field](#)

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

Katakana ( " あいうえお ") returns アイウエオ

## Left

### Purpose

Returns the specified numberOfCharacters in text, counting from the left.

### Format

Left (text ; numberOfCharacters)

### Parameters

text - any [text expression](#) or text [field](#)

numberOfCharacters - any numeric expression or field containing a number

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

Left ("Manufacturing" ; 4) returns **Manu**.

Left (Name ; Position (Name ; " " ; 1 ; 1)) returns **Sophie**, when the Name field contains Sophie Tang.

Left (PostalCode ; 3) & Upper (Left (LastName ; 4)) returns **481JOHN** when the PostalCode field contains 48187 and LastName contains Johnson.

## LeftValues

### Purpose

Returns values contained in text, according to the specified numberOfValues, counting from the left.

### Format

LeftValues (text ; numberOfValues)

### Parameters

text - any [text expression](#) or text [field](#)

numberOfValues - any numeric expression or field containing a number

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

Values are text items separated by carriage returns. A value can be empty, a single character, a word, a sentence, or a paragraph. When you press Return, you start creating a new value. The last value will be recognized with or without a carriage return.

Each returned value ends with a carriage return, allowing lists to be easily concatenated.

### Examples

LeftValues ("Plaid¶Canvas¶Suitcase" ; 2) returns

**Plaid**

**Canvas**

LeftValues (list ; 1) returns

**Sophie**

when the text being evaluated contains

- Sophie
- Bill

## LeftWords

### Purpose

Returns text containing the specified numberOfWords in text, counting from the left.

### Format

LeftWords (text ; numberOfWords)

### Parameters

text - any [text expression](#) or text [field](#)

numberOfWords - any numeric [expression](#) or field containing a number

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

LeftWords ("Plaid Canvas Suitcase" ; 2) returns **Plaid Canvas**.

LeftWords (Name ; 1) returns **Sophie**, when the Name field contains Sophie Tang.

## Length

### Purpose

Returns the number of characters in a field, including all spaces, numbers, and special characters.

### Format

`Length(field)`

### Parameters

`field` - any text, number, date, time, timestamp, or container [field](#), or any [text expression](#) or numeric expression

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

For a container field, `Length` returns the size of the original file in bytes.

### Examples

`Length("John")` returns **4**.

`Length(Description)` returns **12** when the value in Description is Modem for PC.

`Length("M1" & Left(Product;5))` returns **7**, when the Product field contains Canvas Backpack.

## Lower

### Purpose

Returns all letters in specified text as lowercase.

### Format

Lower (text)

### Parameters

text - any [text expression](#) or text [field](#)

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Note

To change how a field displays without modifying its contents, see [Text formatting functions](#).

### Examples

Lower ("ABCD") returns **abcd**.

Lower (Course) returns **history**, when the Course field contains History.

Lower ("YOUR BILL IS OVERDUE") returns **your bill is overdue**.



## Middle

### Purpose

Extracts the specified numberOfCharacters from text, starting at the specified character position.

### Format

Middle(text;start;numberOfCharacters)

### Parameters

text - any [text expression](#) or text [field](#)

start - any numeric expression or field containing a number

numberOfCharacters - any numeric expression or field containing a number

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

Middle("(408)555-9054";2;3) returns **408**.

Middle(PhoneNumber;2;3) returns **408** when the PhoneNumber field contains (408) 555-9054.

Middle("abcdefghij";5;2) returns **ef**.

Middle(Name;Position(Name;" ";1;1)+1;3) returns **Smi**, when the text field Name contains John Smith.

## MiddleValues

### Purpose

Returns text containing the specified numberOfValues in text, starting at startingValue.

### Format

MiddleValues (text ; startingValue ; numberOfValues)

### Parameters

text - any [text expression](#) or text [field](#)

startingValue - any numeric expression or field containing a number

numberOfValues - any numeric expression or field containing a number

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

Values are text items separated by carriage returns. A value can be empty, a single character, a word, a sentence or a paragraph. When you press Return you start creating a new value. The last value will be recognized with or without a carriage return.

Each value that is returned ends with a carriage return, allowing lists to be easily concatenated.

### Examples

MiddleValues ("Plaid¶Canvas¶Suitcase" ; 2 ; 1) returns

#### Canvas

MiddleValues (list ; 2 ; 2) returns

#### Bill

#### John

when the list field contains

- Sophie
- Bill
- John

## MiddleWords

### Purpose

Returns text containing the specified numberOfWords from specified text, starting at startingWord.

### Format

MiddleWords (text ; startingWord ; numberOfWords)

### Parameters

text - any [text expression](#) or text [field](#)

startingWord - any numeric expression or field containing a number

numberOfWords - any numeric expression or field containing a number

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

MiddleWords ("Plaid, Canvas, Suitcase"; 2; 2) returns **Canvas, Suitcase**.

MiddleWords (Name; 1; 2) returns **Brigitte Erika**, when the Name field contains Brigitte Erika Durand.

## NumToJText

### Purpose

Converts Arabic numerals to Japanese text.

### Format

NumToJText (number ; separator ; characterType)

### Parameters

number - any numeric [expression](#) or [field](#) containing a number

separator - a number from 0 to 3 representing a separator

characterType - a number from 0 to 3 representing a type

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

If the value for `separator` and `characterType` are blank or other than 0 to 3, then 0 is used.

#### Separator:

- 0 - no separator
- 1 - every 3 digits (thousands)
- 2 - ten thousands( 万 ) and millions( 億 ) unit
- 3 - tens( 十 ), hundreds( 百 ), thousands( 千 ), ten thousands( 万 ) and millions( 億 ) unit

#### Type:

- 0 - half width (Hankaku) number
- 1 - full width (Zenkaku) number
- 2 - Kanji character number 一 二 三
- 3 - Traditional-old-style Kanji character number 壹 貳 參

### Examples

NumToJText (123456789 ; 2 ; 0) returns 1億2345万6789

NumToJText (123456789 ; 3 ; 2) returns 一億二千三百四十五万六千七百八十九

## PatternCount

### Purpose

Returns the number of occurrences of searchString in text.

### Format

`PatternCount (text ; searchString)`

### Parameters

text - any [text expression](#) or text [field](#)

searchString - any text expression or text field representing the set of characters you want to find

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

This function is not case-sensitive.

### Examples

`PatternCount ("Mississippi" ; "is")` returns **2**.

`PatternCount ("Mississippi" ; "issi")` returns **1** (the function isn't inclusive).

`PatternCount (Attending ; "Guest")` returns **1** if the Guest checkbox is one of the items selected in the Attending field.

`PatternCount (Get (AccountPrivilegeSetName) ; "Guest")` returns **0** if the current account has the Full Access privilege set.

## Position

### Purpose

Returns the starting position of the specified occurrence of searchString in text.

### Format

`Position(text;searchString;start;occurrence)`

### Parameters

`text` - any [text expression](#) or text [field](#)

`searchString` - any text expression or text field representing the set of characters you want to find.

`start` - any numeric expression, or field containing a number, representing the number of characters from the start of the text string at which to begin the search. A start value of 1 or less begins the search from the first character of the text string.

`occurrence` - any numeric expression or field containing a number, representing which instance of the text string you want to find. A negative occurrence value causes the scan to go in the opposite direction from start. A zero value for occurrence is invalid and returns a result of zero.

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

This function is not case-sensitive. If `searchString` isn't contained in `text` or if there was no specified occurrence, zero is returned.

### Examples

`Position("Mississippi";"iss";1;1)` returns **2**.

`Position("Mississippi";"iss";1;2)` returns **5**.

`Position("Mississippi";"iss";3;1)` returns **5**.

`Left(Name;Position(Name;" ";1;1)-1)` returns **William**, when `Name` is a text field that contains William Smith.

`Right(Name;Length(Name) - Position(Name;" ";Length(Name);-1))` returns **Smith**.  
Using operators in formulas

## Proper

### Purpose

Returns the first letter of each word in text as uppercase and all other letters as lowercase.

### Format

Proper (text)

### Parameters

text - any [text expression](#) or text [field](#)

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

Proper ("ABCD") returns **Abcd**.

Proper (Name) returns **Yumiko Kitagawa**, when the Name field contains YUMIKO KITAGAWA.

## Quote

### Purpose

Returns the text form of text enclosed in quotation marks.

### Format

Quote (text)

### Parameters

text - any [text expression](#) or [field](#)

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

This function protects text from being evaluated by the `Evaluate` function. Special characters within `text` are escaped appropriately.

### Examples

Quote ("hello") returns **"hello"**.

Quote ("abc\¶") returns **"abc\¶"**.

Quote ("say \"hello\" fred") returns **"say \"hello\" fred"**.

Evaluate (Quote ("1 + 2")) returns **1 + 2**.

Evaluate ("1 + 2" & Quote (" - 1 + 2")) returns **3 - 1 + 2**.



## Replace

### Purpose

Replaces a string of characters in specified text with replacementText.

### Format

Replace (text ; start ; numberOfCharacters ; replacementText)

### Parameters

text - any [text expression](#) or text [field](#)

start - any numeric expression or field containing a number representing the starting position in text

numberOfCharacters - any numeric expression or field containing a number representing the number of characters to remove from text

replacementText - any text expression or field containing the text to replace in the original string

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Character replacement in text begins at the start character position and continues for numberOfCharacters characters. Compare to the Substitute function.

### Examples

Replace ("1234567" ; 5 ; 1 ; "X") returns **1234X67**.

Replace ("1234567" ; 5 ; 1 ; "XX") returns **1234XX67**.

Replace ("1234567" ; 5 ; 2 ; "X") returns **1234X7**.

Replace ("William" ; 3 ; 4 ; "NEW TEXT") returns **WiNEW TEXTm**.

Replace (PhoneNumber ; 1 ; 3 ; "415") returns **415-555-9054**, when the PhoneNumber field contains 408-555-9054.

## Right

### Purpose

Returns the specified numberOfCharacters in text, counting from the right.

### Format

Right (text ; numberOfCharacters)

### Parameters

text - any [text expression](#) or text [field](#)

numberOfCharacters - any numeric expression or field containing a number

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

Right ("Manufacturing" ; 4) returns **ring**.

Right (Name ; Length (Name) - Position (Name ; " " ; 1 ; 1)) returns **Cannon**, when the Name field contains Michelle Cannon.

Right (SerialNumber ; 3) & Upper (Left (LastName ; 4)) returns **187FERR** when the SerialNumber text field contains 00-48-187 and LastName contains Ferrini.

## RightValues

### Purpose

Returns values contained in text, according to the specified numberOfValues, counting from the right.

### Format

RightValues (text ; numberOfValues)

### Parameters

text - any [text expression](#) or text [field](#)

numberOfValues - any numeric expression or field containing a number

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

Values are text items separated by carriage returns. You can place several items together to create a carriage return-delimited list of values. A value can be empty, a single character, a word, a sentence, or a paragraph. When you press Return you start creating a new value. The last value will be recognized with or without a carriage return.

When the text parameter is a literal string as in the example below, you must insert a literal carriage return character between each item in the list. In the Specify Calculation dialog box, click the ¶ button to insert a literal carriage return character.

Each value that is returned ends with a carriage return, allowing lists to be easily concatenated.

### Examples

RightValues ("Plaid¶Canvas¶Suitcase" ; 2) returns

**Canvas**

**Suitcase**

RightValues (names ; 1) returns

**John**

when the names field contains

- Sophie
- Bill
- John

## RightWords

### Purpose

Returns text containing the specified numberOfWords in text, counting from the right.

### Format

RightWords (text ; numberOfWords)

### Parameters

text - any [text expression](#) or text [field](#)

numberOfWords - any numeric expression or field containing a number

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

RightWords ("Plaid Canvas Suitcase";2) returns **Canvas Suitcase**.

RightWords (Name;1) returns **Virtanen**, when the Name field contains Matti Virtanen.

## RomanHankaku

### Purpose

Converts from Zenkaku alphanumeric and symbols to Hankaku alphanumeric and symbols.

### Format

RomanHankaku (text)

### Parameters

text - any [text expression](#) or text [field](#)

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

RomanHankaku ("M a c i n t o s h") returns **Macintosh**.

## RomanZenkaku

### Purpose

Converts from Hankaku alphanumeric and symbols to Zenkaku alphanumeric and symbols.

### Format

RomanZenkaku (text)

### Parameters

text - any [text expression](#) or text [field](#)

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

RomanZenkaku (“Macintosh”) returns **M a c i n t o s h**.

## SerialIncrement

### Purpose

Returns specified text and numbers, with the numbers incremented by the specified amount.

### Format

```
SerialIncrement (text ; incrementBy)
```

### Parameters

`text` - any text that also contains a number

`incrementBy` - any numeric [expression](#) to increment the text by

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

This function doesn't remove the text in `text`, which normally happens when performing standard math against a value that contains text.

If the `incrementBy` value is a decimal number, then only the integer portion of `incrementBy` value is added to the last number in `text`. Any character other than a number is considered a separator. You can use both positive and negative `incrementBy` values.

### Examples

```
SerialIncrement ("abc12" ; 1) returns abc13.
```

```
SerialIncrement ("abc12" ; 7) returns abc19.
```

```
SerialIncrement ("abc12" ; -1) returns abc11.
```

```
SerialIncrement ("abc12" ; 1.2) returns abc13.
```

```
SerialIncrement ("abc1.2" ; 1.2) returns abc1.3.
```

In the example below any character other than a number is considered as a separator and the number on the far right is incremented.

```
SerialIncrement ("abc123;999" ; 1) returns abc123;1000.
```

## Substitute

### Purpose

Returns a text string with every occurrence of `searchString` in specified text replaced by `replaceString`.

### Format

`Substitute(text;searchString;replaceString)`

### Parameters

`text` - any [text expression](#) or text [field](#)

`searchString` - any text expression or text field

`replaceString` - any text expression or text field

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

This function is case-sensitive. Compare to the `Replace` function.

Multiple substitutions are allowed when you enclose each pair of `searchString` and `replaceString` parameters within brackets `[]` and separate them with semicolons. FileMaker supports up to 999 nested substitute conditions. Each search and replace list item is also separated by semicolons. For example:

```
Substitute(text; [search1; replace1]; [search2; replace2]; ... [searchN;
replaceN])
```

### Examples

`Substitute(Description;"WYSIWYG."; "What you see is what you get.")` replaces every occurrence of the acronym "WYSIWYG." in the `Description` field with the phrase **What you see is what you get.**

`Substitute(text; ["a"; "A"]; ["b"; "B"])` replaces every lowercase a or b with **A** or **B**.



## Trim

### Purpose

Returns text stripped of all leading and trailing spaces.

### Format

Trim(text)

### Parameters

text - any [text expression](#) or text [field](#)

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Use this function to remove unneeded spaces when you convert files from other programs or systems that require a fixed number of characters per field, or to remove spaces accidentally typed during data entry.

### Examples

Trim(" Tom ") returns **Tom**.

Trim(Middle("00230013 William 1234";9;9)) returns **William**.

## TrimAll

### Purpose

Returns a copy of text with specified spaces (such as spaces between text, or non-Roman spaces such as full- and half-width) removed or inserted.

### Format

```
TrimAll (text ; trimSpaces ; trimType)
```

### Parameters

text - any [text expression](#) or text [field](#)

trimSpaces - 0 or False, 1 or True

trimType - 0 through 3 depending on the trim style that you wish to use

### Data type returned

text

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Use TrimAll to work with spaces between text or non-Roman spaces such as full- and half-width spaces; otherwise, use the [Trim function](#).

Set trimSpaces to True (1) if you want to remove full-width spaces. Set trimSpaces to False (0) if you want to keep full-width spaces.

**Note** Full-width spaces are only present in some non-Roman languages like Japanese. If you only use Roman languages, set trimSpaces to False (0).

A character is considered Roman if its [Unicode](#) value is less than U+2F00. Any character whose Unicode value is greater than or equal to U+2F00 is considered non-Roman.

Characters within the Roman range are those belonging to the following character blocks: Latin, Latin-1 Supplement, Latin Extended-A & B, IPA Extensions, Spacing Modifier Letters, Combining Diacritical Marks, Greek, Cyrillic, Armenian, Hebrew, Arabic, Devanagari, Bengali, Gurmukhi, Gujarati, Oriya, Tamil, Telugu, Kannada, Malayalam, Thai, Lao, Tibetan, Georgian, Hangul Jamo, and additional Latin and Greek extended blocks.

Symbols within the Roman range include punctuation characters, superscripts, sub-scripts, currency symbols, combining marks for symbols, letter-like symbols, number forms, arrows, math operators, control pictures, geometric shapes, dingbats, and so on.

Characters within the non-Roman range are those belonging to the CJK symbols/punctuations area, Hiragana, Katakana, Bopomofo, Hangul compatibility Jamo, Kanbun, CJK unified ideographs, and so on.

Spaces are removed or inserted depending on the value of `trimType`, as given in the following tables:

| This trimType value | Does this  |
|---------------------|--|
| 0                   | Removes all spaces between non-Roman and Roman characters (always leave one space between Roman words).  |
| 1                   | Always includes a half-width space between non-Roman and Roman characters (always leave one space between Roman words).  |
| 2                   | Removes spaces between non-Roman characters (reduce multiple space between non-Roman and Roman words to 1 space; do not add spaces if there are none; always leave one space between Roman words). |
| 3                   | Removes all spaces everywhere.   |

In all cases, spaces between non-Roman characters are removed.

| Type | Non-Roman - Non-Roman | Non-Roman - Roman | Roman - Roman |
|------|-----------------------|-------------------|---------------|
| 0    | Remove                | Remove            | 1 space       |
| 1    | Remove                | 1 space*          | 1 space       |
| 2    | Remove                | 1 space           | 1 space       |
| 3    | Remove                | Remove            | Remove        |

\* = insert space between non-Roman and Roman text if there isn't one.

## Examples

TrimAll(" Julian Scott Dunn ";0;0) returns **Julian Scott Dunn**.

TrimAll(名前,1,0) returns 山田太郎 if the value of 名前 field is 山田 太郎

TrimAll("ファイルメーカーPro は高品質",1,0) returns ファイルメーカーProは高品質

# Upper

## Purpose

Returns all letters in specified text as uppercase.

## Format

Upper (text)

## Parameters

text - any [text expression](#) or text [field](#)

## Data type returned

text

## Originated in

FileMaker Pro 6.0 or earlier

## Description

Use the Upper function to ensure consistent data entry of such things as state abbreviations or postal codes.

**Note** To change how a field displays without modifying its contents, see [Text formatting functions](#).

## Examples

Upper ("Ca") returns **CA**.

Upper ("12n34p") returns **12N34P**.

## ValueCount

### Purpose

Returns a count of the total number of values in specified text.

### Format

ValueCount (text)

### Parameters

text - any [text expression](#) or text [field](#)

---

**Important** See [Design functions](#) for information about literal text parameters.

---

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

Values are text items separated by carriage returns. You can place several items together to create a carriage-return-delimited list of values. A value can be empty, a single character, a word, a sentence, or a paragraph. When you press Return you start creating a new value. The last value will be recognized with or without a trailing carriage return.

**Note** If the last character in `text` is a carriage return, ValueCount does not recognize that carriage return as a value.

When the `text` parameter is a literal string as in the example below, you must insert a literal carriage return character between each item in the list. In the Specify Calculation dialog box, click the ¶ button to insert a literal carriage return character.

### Examples

ValueCount ("Item 1¶Item 2¶Item 3") returns **3**.

ValueCount (ValueListItems ("Employees"; "Employee Names")) returns the total number of values in the Employee Names value list in the Employees [database file](#).

## WordCount

### Purpose

Returns a count of the total number of words in specified text.

### Format

WordCount (text)

### Parameters

text - any [text expression](#) or text [field](#)

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

WordCount ("The sun is rising.") returns **4**.

WordCount (Letter) returns the total number of words in the Letter field. If the Letter field contains the text x=y=1.5, this example returns **3** because each equal (=) character identifies the beginning of a new word.

**Note** In addition to space, tab, enter, and carriage return characters, characters such as equal (=), ampersand (&), hyphen (-), and so on identify the beginning of a new word. For more information about word separators, search the FileMaker Knowledge Base available at <http://help.filemaker.com>.

## Text formatting functions

Text formatting [functions](#) can be used to change the color, font, size, and style of the specified text. For example, you could use the `TextFont` function to change the font of the specified text from Arial to Courier. You can use these functions together to change the appearance of text on your layouts.

Text formatting functions operate on these parameters:

- fields of type text
- text [constants](#) (in quotations)
- [expressions](#) having a text result

Click a function name for details.

| This function                    | Does this  |
|----------------------------------|--|
| <a href="#">RGB</a>              | Returns an integer number from 0 to 16777215, obtained by combining color values for red, green, and blue. |
| <a href="#">TextColor</a>        | Changes the color of text to the color specified by the RGB function.                                      |
| <a href="#">TextColorRemove</a>  | Removes all font colors in specified text, or removes the font color specified by the RGB function.        |
| <a href="#">TextFont</a>         | Changes the font of text to the specified <code>fontName</code> .  |
| <a href="#">TextFontRemove</a>   | Removes all fonts or the <code>fontToRemove</code> from specified text.                                    |
| <a href="#">TextFormatRemove</a> | Removes all text formatting from specified text in a single action.  |
| <a href="#">TextSize</a>         | Changes the font size of specified text to <code>fontSize</code> .   |
| <a href="#">TextSizeRemove</a>   | Removes all font sizes in text or the <code>fontSizeToRemove</code> .                                      |
| <a href="#">TextStyleAdd</a>     | Adds the specified styles to text in a single action.  |
| <a href="#">TextStyleRemove</a>  | Removes the specified styles from text in a single action.   |

## RGB

### Purpose

Returns an integer number from 0 to 16777215, obtained by combining color values for red, green, and blue.

### Format

RGB (red;green;blue)

### Parameters

red - any numeric [expression](#) or numeric [field](#) containing a value from 0 to 255

green - any numeric expression or numeric field containing a value from 0 to 255

blue - any numeric expression or numeric field containing a value from 0 to 255

### Data type returned

number

### Originated in

FileMaker Pro 7.0

### Description

Numbers returned by this function can be passed as the color parameter in the `TextColor` or `TextColorRemove` functions. The RGB function uses the following formula to calculate the result:

$$\text{red} * 256^2 + \text{green} * 256 + \text{blue}$$

where  $256^2 = 65536$

**Tip** To determine the RGB value of a color, in Layout mode, click the Fill color palette in the [formatting bar](#) and choose **Other Color**. Values are shown for each of the basic colors. In OS X, select the Color Sliders tab. Choose **RGB Sliders** from the list.

### Examples

RGB (255;0;0) returns **16711680** representing red.

RGB (0;255;0) returns **65280** representing green.

RGB (0;0;255) returns **255** representing blue.

RGB (0;0;0) returns **0** representing black.

RGB (255;255;255) returns **16777215** representing white.

Using a table with text fields `FirstName` and `LastName`, specify the following auto-enter calculation for a third field called `FullName` that displays `FirstName` in orange and `LastName` in purple:

```
TextColor (FirstName;RGB (255;165;0)) &" "&
TextColor (LastName;RGB (160;32;240))
```



## TextColor

### Purpose

Changes the color of text to the color specified by the RGB function.

### Format

`TextColor (text ; RGB (red ; green ; blue )`

### Parameters

`text` - any [text expression](#) or [text field](#)

`RGB (red ; green ; blue)` - any integer from 0 to 16777215 obtained by combining the `red`, `green`, and `blue` values (each ranging from 0 to 255) to represent a color

### Data type returned

`text`

### Originated in

FileMaker Pro 7.0

### Description

Use this function to change the color of `text`.

**Note** Text formatting options will be lost if the field type that is returned is something other than text.

**Tip** To determine the RGB value of a color, in Layout mode, click the Fill color palette in the [formatting bar](#) and choose **Other Color**. Values are shown for each of the basic colors. In OS X, select the Color Sliders tab. Choose **RGB Sliders** from the list.

### Examples

`TextColor ("Plaid" ; RGB (255 ; 0 ; 0) )` returns the word **Plaid** in red.

`TextColor ("Plaid" ; RGB (0 ; 255 ; 0) )` returns the word **Plaid** in green.

`TextColor ("Plaid" ; RGB (0 ; 0 ; 255) )` returns the word **Plaid** in blue.

`TextColor ("Plaid" ; RGB (0 ; 0 ; 0) )` returns the word **Plaid** in black.

`TextSize ( TextFont ( TextColor ( MyTable::MyText ; RGB ( 0 ; 125 ; 125 ) ) ; "Courier" ) ; 12)` returns the text contained in **MyTable::MyText** formatted as 12pt. green text with the Courier font.

## TextColorRemove

### Purpose

Removes all font colors in specified text, or removes the font color specified by the RGB function.

### Format

```
TextColorRemove (text { ;RGB (red;green;blue) })
```

### Parameters

`text` - any [text expression](#) or text [field](#).

`RGB (red;green;blue)` - any integer number from 0 to 16777215 obtained by combining the red, green, and blue values (each ranging from 0 to 255) to represent a color.

Parameters in braces { } are optional.

### Data type returned

text

### Originated in

FileMaker Pro 8.0

### Description

Use this function to revert `text` to the default font color for the field. If you don't use the `RGB` function to specify a color, all of the text displays in the default font color that was set in Layout mode for the field. When the font color is specified by the `RGB` function, only the specified font color is removed from every portion of the text displayed in that color and these same portions of the text are then displayed in the field's default font color.

**Note** Text formatting options will be lost if the field type that is returned is something other than text.

### Examples

`TextColorRemove ("Red Text and Green Text")` returns **Red Text and Green Text** displayed in the field's default font color.

`TextColorRemove ("Red Text and Green Text";RGB(255;0;0))` returns **Red Text and Green Text** with only the pure red font color removed from the words **Red Text**.

## TextFont

### Purpose

Changes the font of text to the specified fontName.

### Format

TextFont (text ; fontName)

### Parameters

text - any [text expression](#) or text [field](#).

fontName - any font name expressed in text.

**Note** TextFont no longer uses the fontScript parameter.

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

Spellings for font names must be correct. Text formatting options will be lost if the field type that is returned is something other than text.

### Examples

TextFont ("Plaid" ; "Courier") returns the word **Plaid** in the Courier font.

TextFont ("Plaid" ; "Arial") returns the word **Plaid** in the Arial font.

TextSize ( TextFont ( TextColor ( MyTable::MyText ; RGB ( 0 ; 125 ; 125 ) ) ; "Courier" ) ; 12) returns the text contained in **MyTable::MyText** formatted as 12pt. green text with the Courier font.

## TextFontRemove

### Purpose

Removes all fonts or the fontToRemove from specified text.

### Format

TextFontRemove (text { ; fontToRemove } )

### Parameters

text - any [text expression](#) or text [field](#).

fontToRemove - any font name expressed in text.

Parameters in braces { } are optional.

**Note** TextFontRemove no longer uses the fontScript parameter.

### Data type returned

text

### Originated in

FileMaker Pro 8.0

### Description

Use this function to revert text to the default for the field. If you don't specify a font, all of the text displays in the default font that was set in Layout mode for the field. When the font is specified by fontToRemove, only the specified font is removed from every portion of the text displayed in that font and these same portions of the text are then displayed in the field's default font.

Spellings for font names must be correct. Text formatting options will be lost if the field type that is returned is something other than text.

### Examples

TextFontRemove("Arial Text and Courier Text") returns **Arial Text and Courier Text** displayed in the field's default font.

TextFontRemove("Arial Text and Courier Text"; "Arial") returns **Arial Text and Courier Text** with the Arial font removed from the words **Arial Text**.

## TextFormatRemove

### Purpose

Removes all text formatting from specified text in a single action.

### Format

TextFormatRemove (text)

### Parameters

text - any [text expression](#) or text [field](#)

### Data type returned

text

### Originated in

FileMaker Pro 8.0

### Description

Use this function to remove all fonts, styles, font sizes, and font colors from the specified text.

### Examples

TextFormatRemove ("Plaid") returns the word **Plaid** without any text formatting applied.

## TextSize

### Purpose

Changes the font size of specified text to `fontSize`.

### Format

```
TextSize(text;fontSize)
```

### Parameters

`text` - any [text expression](#) or text or number [field](#)

`fontSize` - any font size expressed as an integer

### Data type returned

text, number

### Originated in

FileMaker Pro 7.0

### Description

The font size is described in points (72 points to the inch). Text formatting options will be lost if the data type that is returned is something other than text or number.

### Examples

`TextSize("Plaid";18)` returns the word **Plaid** in 18 point text.

`TextSize("Plaid";24)` returns the word **Plaid** in 24 point text.

`TextSize( TextFont( TextColor( MyTable::MyText; RGB( 0 ; 125 ; 125 ) ); "Courier" ) ; 12)` returns the text contained in **MyTable::MyText** formatted as 12pt. green text with the Courier font.

## TextSizeRemove

### Purpose

Removes all font sizes in text or the font sizeToRemove.

### Format

```
TextSizeRemove (text { ;sizeToRemove } )
```

### Parameters

text - any [text expression](#) or text [field](#).

sizeToRemove - any font size expressed as an integer.

Parameters in braces { } are optional.

### Data type returned

text

### Originated in

FileMaker Pro 8.0

### Description

Use this function to revert `text` to the default font size for the field. If you don't specify a size, all of the text displays in the default font size that was set in Layout mode for the field. When the font size is specified by `sizeToRemove`, only the specified font size is removed from every portion of the text displayed in that size and these same portions of the text are then displayed in the field's default font size.

The font size is described in points (72 points to the inch). Text formatting options will be lost if the field type that is returned is something other than text.

### Examples

`TextSizeRemove ("10 Point Text and 18 Point Text")` returns **10 Point Text and 18 Point Text** displayed in the field's default font size.

`TextSizeRemove ("10 Point Text and 18 Point Text";18)` returns **10 Point Text and 18 Point Text** with the 18 point font size removed from the words **18 Point Text**.

## TextStyleAdd

### Purpose

Adds the specified styles to text in a single action.

### Format

```
TextStyleAdd(text; styles)
```

### Parameters

`text` - any [text expression](#) or text [field](#)

`styles` - any named style listed below in Description

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

You can add multiple styles by using the + operator between style names. Negative values are not valid. All styles will be removed, if the only style specified is Plain. Plain is ignored if mixed with other styles. Styles are not case-sensitive and do not contain spaces.

Text formatting options will be lost if the field type that is returned is something other than text.

The styles that are available are:

- Plain
- Bold
- Italic
- Underline
- HighlightYellow
- Condense
- Extend
- Strikethrough
- SmallCaps
- Superscript
- Subscript
- Uppercase
- Lowercase
- Titlecase
- WordUnderline
- DoubleUnderline
- AllStyles (all available styles)



**Note** To format or change the case of text, use the `Lower`, `Upper`, or `Proper` function (see [Text functions](#)).

## Examples

`TextStyleAdd("Plaid";Italic)` returns the word **Plaid** in italics.

`TextStyleAdd(FirstName;Bold+Underline)` returns **Sophie** in bold, underlined text when the `FirstName` field contains `Sophie`.

The following calculation removes all styles from the text, then italicizes the entire phrase.

```
TextStyleAdd(TextStyleAdd(FirstName;Plain);Italic)
```

The following calculation creates two descriptions of styles, then concatenates two phrases using these styles. Using the `Let` function is an effective way to avoid creating a long and complex `TextStyleAdd` statement.

```
Let ([TitleStyle=Smallcaps+Titlecase;BodyStyle=Plain];
TextStyleAdd(titleField;titleStyle)&"¶¶" &
TextStyleAdd(bodyField;BodyStyle))
```

In the following example, to find every occurrence of several words and change their style, use the `Substitute` function combined with the `TextStyleAdd` function.

```
Substitute(ArticleBody;["Phrase1";TextStyleAdd("Phrase 1";Italic)];["Phrase 2";TextStyleAdd("Phrase 2";Bold)];)
```

## TextStyleRemove

### Purpose

Removes the specified styles from text in a single action.

### Format

`TextStyleRemove (text ; styles)`

### Parameters

`text` - any [text expression](#) or text [field](#)

`styles` - any named style from the list of available styles

### Data type returned

text

### Originated in

FileMaker Pro 7.0

### Description

You can remove multiple styles by using the + operator between style names. Negative values are not valid. The Plain styles cannot be used for this function. Plain is ignored if intermingled with other styles. Styles are not case-sensitive and do not contain spaces.

An additional style called AllStyles has been provided to make it easier to remove all styles. Text formatting options will be lost if the field type that is returned is something other than text.

The styles that are available are:

- Bold
- Italic
- Underline
- HighlightYellow
- Condense
- Extend
- Strikethrough
- SmallCaps
- Superscript
- Subscript
- Uppercase
- Lowercase
- Titlecase
- WordUnderline
- DoubleUnderline
- AllStyles (all available styles)

## Examples

`TextStyleRemove("Plaid";Italic)` returns the word **Plaid** with the italics style removed.

`TextStyleRemove(FirstName;Bold + Underline)` returns **Sophie** with the bold and underlined styles removed when the `FirstName` field contains Sophie.

`TextStyleRemove(FirstName;AllStyles)` returns **Sophie** without any styles.

`TextStyleRemove(MyTable::MyText;HighlightYellow)` returns the text contained in **MyTable::MyText** with the `HighlightYellow` style removed.

## Time functions

Time [functions](#) calculate times and manipulate time information.

Click a function name for details.

| This function           | Returns   |
|-------------------------|---|
| <a href="#">Hour</a>    | A number representing the hour portion (0-23) of a specified time.      |
| <a href="#">Minute</a>  | A number representing the minute portion (0-59) of a specified time.    |
| <a href="#">Seconds</a> | A number representing the seconds portion (0-59) of a specified time.   |
| <a href="#">Time</a>    | A time result with the specified number of hours, minutes, and seconds. |

**Note** You can also use the [Timestamp function](#) for time information.

## Hour

### Purpose

Returns a number representing the hour portion (0-23) of a specified time.

### Format

Hour (time)

### Parameters

time - any time value or [field](#) of type time

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

Hour ("12:15:23") returns **12**.

Hour(Duration) + (Minute(Duration)/60) returns **2.5**, when the Duration time field contains 2:30:15.

If(Hour(HoursWorked) > 8; "Overtime Pay"; " ") returns **Overtime Pay** when the number of hours in HoursWorked is greater than 8.

Hour(CheckIn) returns **3** when the value of CheckIn is 3:24.

## Minute

### Purpose

Returns a number representing the minute portion (0-59) of a specified time.

### Format

Minute(time)

### Parameters

time - any time value or [field](#) of type time

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

Minute("12:15:23") returns **15**.

Hour(Duration) + (Minute(Duration)/60) returns **2.5**, if the Duration time field contains 2:30:15.

**Note** If no minute value is specified, 0 is returned.

## Seconds

### Purpose

Returns a number representing the seconds portion (0-59) of a specified time.

### Format

Seconds (time)

### Parameters

time - any time value or [field](#) of type time

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

Seconds ("12:15:23") returns **23**.

**Note** If no seconds value is specified, 0 is returned.

## Time

### Purpose

Returns a time result with the specified number of hours, minutes, and seconds.

### Format

Time (hours ; minutes ; seconds)

### Parameters

hours - the hour value of a time

minutes - the minutes value of a time

seconds - the seconds value of a time

### Data type returned

time

### Originated in

FileMaker Pro 6.0 or earlier

### Description

FileMaker Pro compensates when you supply fractional hours or minutes. The result is the time, formatted according to the time format of the field in the current layout.

Use the Time function or the GetAsTime function to enter a time [constant](#) into a [formula](#).

### Examples

Time (4 ; 14 ; 32) returns **4:14:32**.

Time (4.5 ; 10 ; 30) returns **4:40:30**.

Time (4 ; 15 ; 70) returns **4:16:10**.



## Timestamp functions

Timestamps are used for a wide variety of synchronization purposes, such as marking the exact date and time at which a particular event occurred.

| This function             | Returns   |
|---------------------------|---|
| <a href="#">Timestamp</a> | A timestamp containing a date as a calendar date and a time as a time of day. |

## Timestamp

### Purpose

Returns a timestamp containing a date as a calendar date and a time as a time of day.

### Format

Timestamp (date;time)

### Parameters

date - any calendar date or date [field](#)

time - any time value or time field

### Data type returned

timestamp

### Originated in

FileMaker Pro 7.0

### Description

The format of the result depends on the date and time formats that were in use when the [database file](#) was created. You can change the date and time formats in your operating system.

### Examples

Timestamp (Date (10;11;2014) ;Time (9;10;30) ) returns **10/11/2014 9:10:30 AM.**

Timestamp (Date (10;11;2014) ;Time (13;10;30) ) returns **10/11/2014 1:10:30 PM.**

Timestamp (Date (10;11;2014) ;Time (10;65;5) ) returns **10/11/2014 11:05:05 AM.**

Timestamp (Date (10;35;2014) ;Time (4;5;6) ) returns **11/4/2014 4:05:06 AM.**

## Trigonometric functions

Trigonometric [functions](#) are used to calculate degrees, angles, and other geometric data.

**Note** All trigonometric functions use radians as the unit of measure. Once you have a result, you can convert the radians into degrees using the [Degrees function](#).

Click a function name for details.

| This function           | Does this  |
|-------------------------|--|
| <a href="#">Acos</a>    | Returns the arccosine (Acos), or inverse cosine, of a number.                  |
| <a href="#">Asin</a>    | Returns the arcsine (Asin), or inverse sine, of a number.                      |
| <a href="#">Atan</a>    | Returns the trigonometric arc tangent (Atan), or inverse tangent, of a number. |
| <a href="#">Cos</a>     | Returns the cosine (Cos) of angleInRadians.                                    |
| <a href="#">Degrees</a> | Converts angleInRadians to degrees.  |
| <a href="#">Pi</a>      | Calculates the value of the <a href="#">constant</a> Pi.                       |
| <a href="#">Radians</a> | Converts angleInDegrees to radians.  |
| <a href="#">Sin</a>     | Returns the sine (Sin) of angleInRadians expressed in radians.                 |
| <a href="#">Tan</a>     | Returns the tangent (Tan) of angleInRadians.                                   |

## Acos

### Purpose

Returns the arccosine (Acos), or inverse cosine, of a number.

### Format

Acos (number)

### Parameters

number - any numeric [expression](#) or [field](#) containing a numeric expression in the range -1 to 1

### Data type returned

number

### Originated in

FileMaker Pro 9.0

### Description

The arccosine is the angle whose cosine is number. The returned angle is given in radians in the range 0 (zero) to Pi. The input number parameter must be between -1 and 1.

If you want to convert the result from radians to degrees, multiply it by 180/Pi or use the [Degrees function](#).

### Examples

Acos (-0.5) returns **2.0943951**.

Acos (-0.5) \*180/Pi returns **120**.

Degrees (Acos (-0.5)) returns **120**.

Acos (2.0) returns ? (not a number).

## Asin

### Purpose

Returns the arcsine (Asin), or inverse sine, of a number.

### Format

Asin (number)

### Parameters

number - any numeric [expression](#) or [field](#) containing a numeric expression in the range -1 to 1

### Data type returned

number

### Originated in

FileMaker Pro 9.0

### Description

The arcsine is the angle whose sine is `number`. The returned angle is given in radians in the range  $-\pi/2$  to  $\pi/2$ . The input number parameter must be between -1 and 1.

To express the arcsine in degrees, multiply the result by  $180/\pi$  or use the [Degrees function](#).

### Examples

Asin(-0.5) returns **-0.523598776**.

Asin(-0.5)\*180/Pi returns **-30**.

Degrees(Asin(-0.5)) returns **-30**.

Asin(2) returns ? (not a number).

## Atan

### Purpose

Returns the trigonometric arc tangent (Atan), or inverse tangent, of a number.

### Format

Atan (number)

### Parameters

number - any numeric [expression](#) or [field](#) containing a numeric expression

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

The arc tangent is the angle, in radians, whose tangent is equal to the specified number.

### Examples

Atan (1) returns **.78539816....**

Degrees (Atan (1) ) returns **45**.

## Cos

### Purpose

Returns the cosine (Cos) of angleInRadians.

### Format

`Cos (angleInRadians)`

### Parameters

angleInRadians - any numeric [expression](#) or [field](#) containing a numeric expression, in radians

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

`Cos (1.047)` returns **.50017107....**

`Cos (Radians (60))` returns **.5**.

## Degrees

### Purpose

Converts angleInRadians to degrees.

### Format

Degrees (angleInRadians)

### Parameters

angleInRadians - any numeric [expression](#) or [field](#) containing a numeric expression, in radians

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Use this function to translate results of trigonometric functions from radians to degrees.

$$\text{Degrees} = \frac{180 \cdot \text{angleInRadians}}{\pi}$$

### Examples

Degrees (Atan (1)) returns **45**.

Degrees (1.0472) returns **60.00014030....**



## Pi

### Purpose

Calculates the value of the constant Pi.

### Format

Pi

### Parameters

None

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

This function calculates the value of the constant Pi ( $\pi$ ), which is approximately 3.14159.

### Examples

Pi \* 15 returns **47.124**.

## Radians

### Purpose

Converts angleInDegrees to radians.

### Format

Radians (angleInDegrees)

### Parameters

angleInDegrees - any numeric [expression](#) or [field](#) containing a numeric expression, in degrees

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

The parameters for FileMaker Pro trigonometric functions must be expressed in radians. If the values you want to use as parameters in a trigonometric equation are in degrees, use this function to convert them to radians first. A degree is equal to Pi/180 radians.

$$\text{Radians} = \frac{\pi \cdot \text{angleInDegrees}}{180}$$

### Examples

Radians(45) returns **.78539816....**

Sin(Radians(30)) returns **.5**.

## Sin

### Purpose

Returns the sine (Sin) of angleInRadians expressed in radians.

### Format

`Sin (angleInRadians)`

### Parameters

angleInRadians - any numeric [expression](#) or [field](#) containing a numeric expression, in radians

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Examples

`Sin (Radians (60))` returns **.86602**.

`Sin (.610865)` returns **.57357624....**

## Tan

### Purpose

Returns the tangent (Tan) of angleInRadians.

### Format

Tan (angleInRadians)

### Parameters

angleInRadians - any numeric [expression](#) or [field](#) containing a numeric expression, in radians

### Data type returned

number

### Originated in

FileMaker Pro 6.0 or earlier

### Description

Use this function to calculate the Tan of angleInRadians.

**Note** With the Tan function, you cannot use values exactly equal to 90 degrees (Pi/2 radians), or multiples of 90 degrees.

$$\text{Tan} = \frac{\text{Sin}(\text{angleInRadians})}{\text{Cos}(\text{angleInRadians})}$$

### Examples

Tan (.13) returns **.13073731....**

Tan (Radians (34) ) returns **.6745085**.

# Glossary

## A

### Access key (Windows)

A key that activates a menu, menu item, or control when used with the ALT key. In Windows, this key corresponds to the underlined letter on a menu, command, or dialog box option.

### Access privileges

Permission to view and work with certain records, fields, layouts, value lists, and scripts and to perform selected activities in a file.

### Account

A user name and (usually) password that accesses a file with a defined level of privileges. There are two predefined accounts: Admin and Guest. Admin is a Full Access account that can be renamed or deleted. At least one Full Access account that is authenticated via FileMaker must be defined for each file. Guest account is a special account that cannot be renamed or deleted, but can be made active or inactive.

### Active pane

In the Script Workspace, the pane (scripts pane, script editing pane, or script steps pane) that's active for use.

### Active tab

In the Script Workspace, the tab in the script editing pane that's active for use.

### ActiveX Automation

A Windows programming and scripting protocol that allows external control of specific commands and actions in FileMaker Pro, including opening and closing FileMaker Pro files, toggling the application's visibility, and performing FileMaker Pro scripts.

### Alpha channel

The alpha channel of an image stores transparency information and controls the transparency of the red, green, and blue color channels.

### Animation

A visual effect that provides feedback while you are performing actions in FileMaker Pro and FileMaker Go, such as switching between slide panels.

### API (Application Programming Interface)

A set of software application building blocks, such as data structures, variables, procedures, and functions, used by programmers.

### AppleScript

A scripting language you can use to control functions of OS X and of applications that support AppleScript (often called *scriptable applications*).

### Apple events

An OS X technology that lets applications communicate with one another. FileMaker Pro can send and receive Apple events to and from applications that support them.

### Ascending sort order

Alphabetical sequence (A to Z) for words, lowest to highest order for numbers, and earliest to latest for dates and times.

**ASCII character set**

American Standard Code for Information Interchange. A standard character set used by computer systems worldwide (often extended for different alphabets).

**Authentication**

The process of checking the validity of an account and password (if one is defined) before assigning privileges and allowing access to a system or a file. An account authenticated via FileMaker Pro or FileMaker Server is referred to as a FileMaker account. (FileMaker Server can also authenticate an account via External Server — an external authentication system such as Apple Open Directory, or a Windows domain.)

**Authorization**

Allowing a file to access a protected file's schema (including its tables, layouts, scripts, and value lists). Such a file is an *authorized file*.

**Auxiliary files**

In a FileMaker Pro Advanced runtime solution, files that are bundled with a primary file.

**B****Badge**

An icon indicating that conditional formatting, script triggers, placeholder text, or tooltips have been applied to a field, object, or layout; that find is available for a field; that an object is a button, popover button, or button bar; or that a solution is a favorite.

**Base directory**

For container fields that store an external reference, the directory in which the referenced data is stored. You can also specify a file system path or a calculation for each open storage container field.

**Binding key**

In FileMaker Pro Advanced, a case-sensitive key from 1 to 24 characters long that links the components of a runtime solution.

**Blank layout**

A layout that contains empty body, header, and footer parts. (In previous versions of FileMaker Pro, a Blank layout was a predefined layout type.)

**Body part**

A layout part that contains individual records from a file.

**Boolean value**

A Boolean value is either true or false. A field containing any number except zero evaluates as true. A field containing zero, no data, or content that does not resolve into a number evaluates as false. For example, a field containing "ABC," "ABC0," or an empty field is false. A field containing "1" or "ABC2" is true.

**Break field**

In a subsummary part, records are grouped (sorted) by values in another field, called the *break field*. Whenever the value of the break field changes, the report "breaks" and FileMaker Pro inserts the subsummary part.

### Browse mode

The FileMaker mode in which you enter and edit information in fields. Groups of fields make up the records of your file. You can view one record at a time (click **Form View** in the layout bar), view your records in a list (click **List View**), or view records arranged in a spreadsheet-like table (click **Table View**).

(Use Browse mode to enter and edit your information; use Layout mode to design how your information is displayed. Use Find mode to find records that match search criteria; use Preview mode to display how your records will print.)

### Button

A layout object that performs a specified script in Browse or Find mode.

### Button bar

A layout object that displays buttons or popover buttons in a horizontal or vertical bar. Button bars contain segments that represent individual buttons consisting of any combination of a button icon and a text label.

## C

### Cache

The amount of memory assigned to FileMaker Pro. A larger cache size increases performance. A smaller cache size saves data to the hard disk more frequently, offering greater protection in case of a system crash.

### Calculation field

A field that returns the result of a calculation of values. You can create a formula for the calculation using functions, constants, operators, and information from other fields in the same record.

### Cascading style sheets

A system of codes or tags that define how a web browser displays information in a webpage. Cascading style sheets provide more control over the layout and appearance of webpages than HTML. Cascading style sheets work like templates for webpages. If a web page contains cascading style sheets, users must view it in a browser that supports cascading style sheets.

### Character encoding

The character set or code page of a file. If necessary, you can specify a character set to be used when importing, exporting, indexing, sorting, and spell-checking files. FileMaker supports ASCII, Windows ANSI, Macintosh, Japanese (Shift-JIS), Unicode UTF-8, Unicode UTF-16, and Unicode UTF-16 Windows.

### Chart

A graphical depiction that makes it easy to compare data and see patterns and trends.

### Chart legend

A key that identifies the colors assigned to a data series in a chart.

### Client

A user that opens a file that is shared on a network, published in a browser, or shared via ODBC/JDBC. FileMaker Network settings and privileges determine how FileMaker Pro and FileMaker Go clients interact with files hosted through FileMaker Pro and FileMaker Server.

### Client application

The application that requests data (using SQL) from a data source (using ODBC or JDBC). Also, FileMaker Pro is a client application when it accesses a file hosted by FileMaker Server.

### Client/server architecture

The relationship between two networked computers that share resources. The client requests services from the server, and the server provides services to the client.

### Clipboard

A temporary storage area in computer memory where FileMaker Pro places the most recent selection you've cut or copied.


### Clone

A copy of a FileMaker Pro file that contains all the field definitions, tables, layouts, scripts, and page setup options, but none of the data.

### Column

When a layout is viewed in Table View, a column corresponds to a field.

### Combo box

A type of list you can set up in Layout mode. In the Data  tab in the Inspector, select **Include arrow to show and hide list**. The list will open only when users click the arrow, not when they enter the field.

### Commit

To save changes to a file. Certain actions such as navigating between records, finding, and sorting do not change the file's modification date. Other actions such as changing data in a record or changing a layout do change the file's modification date.

### Concealed edit box

A field set up to conceal confidential information by displaying dots in place of data.

### Consistency check

A consistency check examines a file that may have been damaged due to, for example, an unexpected quit. FileMaker Pro reads every file block and verifies that the internal structure of the block is valid and the block is properly linked to other blocks in the file. (A consistency check does not read all the data within each file block or check the schema or higher-level structures in the file; these tasks are performed by file recovery.)

### Constant

In a formula, an unchanging value. For example, a constant can be a field name, a text literal ("Total:"), or a number. The value of the constant doesn't change from record to record as the formula is evaluated.

### Container data type

Pictures, movies, and documents such as Microsoft Word files and PDF files can be inserted in a container field. Data in container fields can be embedded in the field, stored by file reference, or stored externally.

### Context

The table occurrence from which calculations and scripts are begun, and from which a relationship is evaluated in the relationships graph.

### Convert

Opening a data file from another application, which creates a new FileMaker Pro file containing the data.

Also refers to opening a file created with a previous version of FileMaker Pro.



**Custom function**

A function that is not one of the default FileMaker Pro functions. In FileMaker Pro Advanced, you can create custom functions that can be reused anywhere in the file.

**Custom menu**

A menu that is not one of the default FileMaker Pro menus. In FileMaker Pro Advanced, you can create custom menus, menu items, and menu sets.

**D****Data Entry Only privilege set**

One of the three predefined privilege sets that appear in every file. The Data Entry Only privilege set allows read/write access to the records in a file, but not design access (for example, the ability to create layouts and value lists).

**Data point**

In a chart, the value of data plotted on the x or y axis displayed as a column, bar, point, bubble, or slice.

**Data series**

Data points that are plotted in a chart. When multiple data series are charted, each data series is displayed in a unique color and is defined in the chart's legend, if included.

**Data source**

A named reference that provides access to another FileMaker file or to an ODBC database. ODBC data sources are also referred to as external SQL sources (ESS).

In charting, the origin of the data to be charted — current found set, current record (delimited data), or related records.

**Data source name (DSN)**

A data structure that contains the information about a specific file that an ODBC driver needs in order to connect to it.

**Data Viewer**

A FileMaker Pro Advanced feature that lets you monitor expressions such as field values, local and global variables, and calculations. You can monitor expressions while running scripts or while testing them in the Script Debugger. You can also monitor field values and variables in the file.

**Database file**

A component of a FileMaker solution.

**Database**

The arrangement of tables and fields and their interrelationships in a file.

**Database Design Report**

A FileMaker Pro Advanced tool that creates a report of your database schema.

**Database Encryption**

A form of encryption that protects “data at rest.” For FileMaker, data at rest is a FileMaker Pro file, and its temporary files, while it is being stored on disk (and not open). An encrypted file is protected from unauthorized access with an encryption password. FileMaker Pro Advanced is required to encrypt files. Encrypted files can be decrypted and re-encrypted as needed.

**Database Management System (DBMS)**

An application that allows users to store, process, and retrieve information in a database.

### Descending sort order

Reverse alphabetical sequence (Z to A) for words, highest to lowest order for numbers, and latest to earliest dates and times.

### Developer Utilities

A FileMaker Pro Advanced feature that lets you bind files into a runtime solution, display files in Kiosk mode, prevent users from modifying the design or structure of databases, and automatically rename sets of files and update links to related files.

### Dialog window

A document window in a modal state. Its behavior is similar to the behavior of a window when a running script is paused. You can create and customize a dialog window using the New Window script step.

### Document window

A standard modeless FileMaker window. You can create and customize a document window using the New Window script step.

### Domain name

The primary subdivision of Internet addresses, which is indicated by the last part of an Internet address after the final period (or dot). In the United States, the standard domains are .com, .edu, .gov, .mil, .org, and .net. In other countries, the top-level domain is usually the country domain.

### Domain name server (DNS)

A server that matches up the URL of a website (for example <http://www.filemaker.com>) with its proper numeric IP address (for example 12.34.56.78).

### Driver

The ODBC or JDBC driver translates SQL queries into commands that a DBMS can understand. It processes ODBC/JDBC calls, submits SQL requests to the data source, and returns the data back to the driver manager, which then routes it to the requesting application (for example, FileMaker Pro).

### Driver manager

The control panel that manages communication between requesting applications and data sources. When an application makes a request via ODBC/JDBC, the driver manager routes the request through the correct driver to the correct data source and returns the data to the requesting application. All ODBC/JDBC drivers and data sources to be used on that computer are registered with the driver manager.

### Drop-down calendar

A field set up to display an interactive monthly calendar when a user enters the field in Browse or Find mode.

### DTD (Document Type Definition)

A formal description of a particular type of XML. It defines a document structure, including the names of data elements and where they may occur within the structure. Valid XML conforms to the rules established in its DTD.

### Dynamic guides

In Layout mode, guidelines that extend horizontally and vertically to help you move, resize, or align objects. Dynamic guides also “snap to” the upper and lower boundaries and the centers of objects as you move, resize, or align the objects.

## E

**Embedded**

For container fields, a file that is stored in the container field. Container data can also be stored by reference or stored externally.

**Encryption password**

The password required to open an encrypted file. For FileMaker Pro, the encryption password is required for the FileMaker Pro user who opens the file, but not for the client who opens a file hosted by FileMaker Pro. FileMaker Pro Advanced is required to enable, disable, or change an encryption password.

**Envelope layout**

A predefined layout with fields arranged for printing on standard business envelopes.

**Ethernet**

A type of fast local area network used for connecting computers and peripherals within the same building or campus.

**Export**

To save data from one file so that it can be used in another file or in another application.

**Expression**

A value or any computation that produces a value. Expressions can contain functions, field values, and constants and can be combined to produce other expressions.

**Extended privilege**

Privileges that determine the data sharing options that are permitted by a privilege set for a file. For example, the Access via FileMaker WebDirect extended privilege determines whether the privilege set allows accessing the file from a web browser.

**External function**

A function written in C or C++ as part of a third-party plug-in that extends the feature set of FileMaker Pro or FileMaker Pro Advanced.

**External script**

A script used by a file, but defined in a different file. Use the Perform Script script step to select a defined script from a related file, or to select a file reference to a file on your hard drive or network.


## F

**Field**

The basic unit of data in a record. You define a field to hold a specific, discrete category of data, such as Last Name, Employee Photo, or to display the result of a calculation. You can define text, number, date, time, timestamp, container, calculation, and summary fields. Field can also refer to the object on a layout that displays the data, such as an edit box, checkbox set, or pop-up menu.

**Field boundary**

In Layout mode, an outline that shows the size of a field. To see field boundaries, choose **View** menu > **Show** > **Field Boundaries**. These boundaries only appear in Layout mode. However,

you can use the Appearance  tab in the Inspector to format objects to have borders that do appear in Browse mode and when you print the layout.

### Field label

Text on a layout or within a field that identifies a field. A field label can be the field name, other text (such as instructions), or the result of a calculation. Field labels can appear in various positions outside the field or as placeholder text within the field. You can change or delete a field label if you want.

### Field name

The name you assign to a field when you define the field. When you place a field onto a layout, you can have FileMaker Pro also place an editable field label that matches the field name. You can also use the field name as placeholder text. Fully qualified fields are displayed in `tablename::fieldname` format.

### Field type

The part of a field definition that determines what kind of data you can enter in the field and the kinds of operations FileMaker Pro can perform with the data. FileMaker Pro can create text, number, date, time, timestamp, container, calculation, and summary fields. (Global fields contain the same value for all records in the file and can be of any type except summary.)

### File path

The location of a file in an operating system as identified by the drive, folders, filename, and file extension.

### FileMaker Go

An application that lets you work with FileMaker Pro solutions (developed in FileMaker Pro) on iPhone, iPod touch, or iPad.

### FileMaker Network

A communications method built into FileMaker Pro that allows you to share files hosted by FileMaker Pro or FileMaker Server with others over a network or in FileMaker Go. The FileMaker Network settings and privileges you set up determine how other users (clients) can open and use the shared file.

### FileMaker WebDirect

A web application for accessing layouts from files in a web browser. Web clients use FileMaker WebDirect to access layouts from files hosted by FileMaker Server.

### Find mode

The FileMaker mode in which you specify criteria for finding a subset of records. (Use Browse mode to enter and edit your information; use Layout mode to design how your information is displayed. Use Find mode to find records that match search criteria; use Preview mode to display how your records will print.)

### Find request

In Find mode, a blank form based on the current layout. Enter search criteria into one or more fields of the find request.

### Firewall

A security system used to prevent unauthorized users from gaining access to a LAN. A firewall usually has a single computer that is connected to the Internet, and all Internet traffic must pass through that computer.

### Floating document window

A standard modeless FileMaker window that stays on top of other windows while users work in FileMaker Pro. You can create and customize a floating window using the New Window script step.

## Focus

Indicates that a record, portal row, tab, option, field, repeating field repetition, layout part, or layout object that has been activated by a click, the Tab key, a navigation button, or a script can then be acted upon in some way. For example, you can change the value in a field that has the focus; you can work with the data that's in a record that has the focus; a calculation function can return the number of the portal row that has the focus.

## Footer part

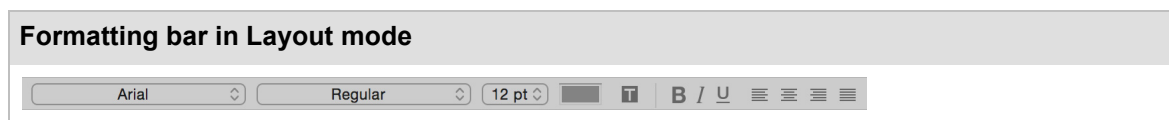
Use the footer part for page numbers or dates. This part appears at the bottom of every screen or page (unless you add a title footer). You can have only one footer in a layout. A field in the footer displays data from the last record on that page.

## Form View

Displays one record at a time. By default, fields appear on separate lines. To select this view, click **Form View** in the layout bar.

## Formatting bar

In Layout and Browse modes, the area that displays options for formatting text and objects. To show or hide the formatting bar, click **Formatting**  in the layout bar.



## Formula

A set of instructions that FileMaker Pro follows to calculate a value used in a field or as the criteria for matching database records.

## Found set

The set of records in a table that are made active when you search for data. When you find all records, the found set is the entire table.

## Full Access privilege set

One of the three predefined privilege sets that appear in every file. The Full Access privilege set allows complete read/write access to a file, including making changes to privileges for the file.

## Fully qualified name

The complete name of a field or layout, expressed using the format `tableName:: [field or layout name]`, where “tableName” is the name of the underlying table occurrence in the relationships graph upon which the field or layout is based. A fully qualified name identifies an exact instance of a field or layout. Because fields and layouts with common names can be based on different tables, FileMaker Pro uses fully qualified names to avoid errors in calculations and scripts.

## Function

A predefined, named formula that performs a specific calculation and returns a single, specific value.

## Function list separator

The punctuation character ; (a semicolon) that separates parameters in a function definition. If you type a comma (,), FileMaker Pro automatically changes it to a semicolon after you close the Specify Calculation dialog box.

## G

### **GIF (Graphics Interchange Format)**

A platform-independent file format often used to distribute graphics on the Internet.

### **Global field**

A field defined with the global storage option can contain one value that's used for all records in a file. Use the value of a global field as a fixed value in calculations, to declare variables in If or Loop script steps, or for fields that rarely need to be updated (for example, a company logo in a container field). A global field can be any field type except summary. A global field can't be indexed.

### **Global variable**

A global variable can be used in a calculation or script anywhere in a file, for example, other scripts or file paths. The value of a global variable is not cleared until the file is closed.

### **Grammar**

A precise description of a formal language, such as XML, consisting of sets of rules for how strings (words) in the language can be generated, and how the strings can be recognized as part of the language.

### **Grand summary**

Total or other aggregate value for all records in the found set.

### **Grand summary part**

Use grand summary parts to view and display summary information (totals, averages, and so on) in summary fields for all records in the found set. You can add one grand summary part at the top (leading) and one grand summary part at the bottom (trailing) of a layout.

### **Grid**

In Layout mode, a series of nonprinting intersecting horizontal and vertical lines. The grid aligns objects you create, resize, move, or position. Objects “snap to” the grid to help you create and edit objects more precisely. Gridlines adjust when you change the unit of measure.

### **Grouped object**

A collection of objects that behaves as one object in Layout mode.

### **Guest**

A user who opens a protected file without specifying an account name and password. The Guest account is assigned a privilege set that determines what guests can do in the file. Guest access may be disabled for a file.

### **Guides**

Nonprinting, movable horizontal and vertical guidelines to help you position and align objects in Layout mode. An object's left or right boundary, top or bottom boundary, or center “snaps to” a guide.

## H

### **Handle**

One of the small squares surrounding a selected object, and used to resize and reshape the object.

## Header part

Use a header part for column headings, titles, and other information that appears only at the top of every page on a layout. FileMaker Pro displays the header in Browse mode and prints it on every page, except the first page if you add a title header. Fields added to a header are printed on every page, using data from the first record on that page.

## Homepage

The starting page for a website. It often has some form of a table of contents that allows viewers to link to other parts of the website.

## Host

After a file has been opened and enabled for sharing, the host is either the first FileMaker Pro user to share the file, or the host is FileMaker Server. Once the host opens the file, other users (clients) can access and change the file. All changes are stored in the file on the computer or device where the file resides. FileMaker Network settings and privileges determine how FileMaker Pro and FileMaker Go clients interact with files hosted through FileMaker Pro or FileMaker Server.

## HTML (Hypertext Markup Language)

A language that is used for displaying and accessing information on the World Wide Web.

## HTTP (Hypertext Transfer Protocol)

The Internet protocol that defines how a web server responds to requests for files.

## Import

To bring (copy) data from a table, another file, or another application into the current table. You can also import scripts from one FileMaker Pro file into another.

## Indexing

An option that can be enabled when defining (or changing) the definition of a field. When indexing is enabled, FileMaker Pro builds a list of all the values that occur in the field in the table. This improves the performance of tasks such as finding data, but it increases the size of the file on disk.

## Inline scroll bar

A scroll bar that appears only while scrolling is occurring.

## Inspector

In Layout mode, a tool that allows you to view and edit the settings for objects, layout parts, and the layout background. You can open multiple inspectors to view and format settings in different tabs at the same time. To open the Inspector, choose **View** menu > **Inspector**. To open another Inspector window, choose **View** menu > **New Inspector**.

## Interactive container

A container field for which the **Interactive content** option in the Inspector is selected. Interactive containers are rendered using the web browser technologies that web viewer uses.

## Internet

An international network of many other networks that are linked using the TCP/IP network protocol.

## Internet service provider (ISP)

The company from which you purchase your connection to the Internet.

**Intranet**

A private TCP/IP network of linked computers within a company or organization.

**IP (Internet Protocol) address**

For IPv4, a four-part number, usually formatted as 12.34.56.78, that uniquely identifies a computer on the Internet; for IPv6, an eight-part number, usually formatted as [2001:0DB8:85A3:08D3:1319:8A2E:0370:7334], that uniquely identifies a computer on the Internet. When referenced in an application, IPv6 addresses must be contained in brackets; for example, [2001:0DB8:85A3:08D3:1319:8A2E:0370:7334].

**J, K****JDBC**

A Java API that uses SQL statements to access data from, and exchange data with, many database management systems. The JDBC driver communicates between your Java applet and the FileMaker Pro or FileMaker Server data source.

**JPEG (Joint Photographic Experts Group)**

A platform-independent file format often used to distribute graphics on the Internet.

**Key**

A column (or columns) that makes a particular row unique (corresponds to a match field).

**Kiosk**

A FileMaker solution that runs full screen, without toolbars or menus. Users click buttons to navigate. In FileMaker Pro Advanced, use the Developer Utilities to create Kiosk solutions. You can bind Kiosk solutions into standalone runtime solutions.

**L****Labels layout**

A predefined layout with fields arranged for printing horizontally or vertically on mailing label stock, and media and index sheets.

**LAN (local area network)**

A connection between computers within a location using cable or a wireless system.

**Layout**

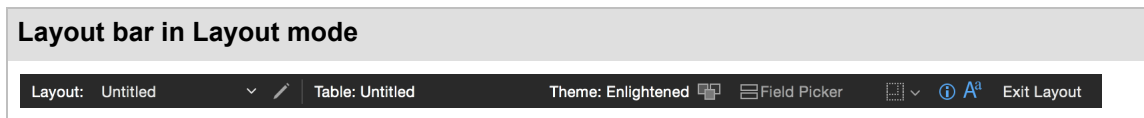
An arrangement of fields, objects, pictures, and layout parts that represents the way information is organized and presented when you browse, preview, or print records. You can design different layouts for entering data, printing reports and mailing labels, displaying webpages, and so on.

**Layout bar**

In the status toolbar, the area that displays options for working with layouts, such as the Layout pop-up menu.







## Layout mode

The FileMaker mode in which you determine how information in fields is presented on the screen and in printed reports.

(Use Browse mode to enter and edit your information; use Layout mode to design how your information is displayed. Use Find mode to find records that match search criteria; use Preview mode to display how your records will print.)

## Layout part

A section of a layout that organizes or summarizes information. Layout parts include body, header, footer, title header, title footer, top navigation, bottom navigation, leading and trailing grand summary, and leading and trailing subsummary.

## Layout pop-up menu

In the layout bar, a pop-up menu from which you can choose **Manage Layouts** (Layout mode) or a defined layout (all modes).

## Layout theme

A collection of coordinated styles that determine the color, object, part, and background attributes, and the fonts used on a layout. Themes also enhance the appearance of a layout or report and give all your layouts a consistent look. A theme does not control the placement or behavior of fields or objects on a layout. A theme is assigned when you create a new layout, but you can change the theme in Layout mode. You can also create a theme by changing the styles used for the layout, and then saving the theme with a new name.

## Layout tools

In the status toolbar in Layout mode, a collection of tools that includes the Selection tool (pointer), Text tool, Line tool, Rectangle tool, Rounded Rectangle tool, Oval tool, Field/Control tool, Button tool, Popover Button tool, Button Bar tool, Tab Control tool, Slide Control tool, Portal tool, Chart tool, Web Viewer tool, Field tool, Part tool, and Format Painter tool.

## Layout types

FileMaker Pro includes several predefined types of layouts and reports to display on different types of devices (such as laptop computer screens or touch device screens), for different purposes (such as browsing records, entering data, or printing reports, mailing labels, or envelopes). After you choose a layout type, you can make additional choices.

To use a predefined layout type, in Layout mode, click **New Layout/Report** in the status toolbar. The assistant guides you through creating the type of layout or report you want. After you finish the assistant, use the tools and commands in Layout mode to tailor the layout for your needs.

## LDAP (lightweight directory access protocol)

A protocol for accessing online directory services.

## Link

On a webpage, text or a graphic which — when you click it — displays an associated webpage or a specific element within a page.

Also, the HTML code that creates a link to another webpage or to a specific element within a page.

### List View

Displays records one record at a time in a list format. To select this view, click **List View** in the layout bar.

### List view layout

A type of Report layout in which fields that you specify appear in columns across the screen or page in one line. Field names are in the header part and the footer part is blank. (In previous versions of FileMaker Pro, a List view layout was a predefined layout type.)

### Locked object

An object on a layout that cannot be edited or deleted. To lock or unlock an object, select it in

Layout mode. In the Inspector, click the **Position**  tab, then choose **Lock** or **Unlock** in the Arrange & Align area. If an object is locked, its selection handles dim.

### Lookup

A lookup matches records and copies data from a related table into a field in the current table. After data is copied, it becomes part of the current table (as well as existing in the table it was copied from). Data copied to a table doesn't automatically change when the values in the related table change.

### Lookup source field

The field in the related table that contains the data you want copied during a lookup.

### Lookup target field

The field that you want data copied to during a lookup.

## M

### Many-to-many relationship

A correspondence between data in database tables in which more than one record in the first table is related to more than one record in another table, and more than one record in that table is related to more than one record in the first table.

### Match field

For relational databases and lookups, a field in a source table and a field in a related table that contain values you want to use to find matching records. (A match field is sometimes called a *key field* or *trigger field*.) In the relationships graph, match fields appear in italics.

For importing records, values in the match fields determine which records in the source table update which records in the target table.

### Menu

A list of menu items. Each menu has a title that appears on the menu bar.

### Menu bar

The area at the top of the window (Windows) or screen (OS X) that displays the installed menu set.

### Menu item

One item listed in a menu on the menu bar. A menu item corresponds with one command, submenu, or separator.

### Menu item properties

All the settings for a menu item, including platform, display title, shortcut, and action.

**Menu set**

The collection of menus that installs on the menu bar.

**Merge field**

A placeholder on a layout for the contents of a field. A merge field expands or contracts in Browse and Preview modes, or when printed, to fit the amount of data in the field for each record.

Merge fields are useful for mail merge form letters; FileMaker Pro uses merge fields in predefined Labels and Envelope layouts.

**Merge variable**

A variable inserted onto a layout that displays values from a local or global variable onto the layout of the current record. You see the merge variable value in Browse, Find, and Preview modes, and when you print records.

**Modal window**

A window that requires user action before users can continue working in FileMaker Pro. Use a modal window to display alert messages or to collect information from users.

**Mode**

In FileMaker Pro, the four different environments (Browse, Find, Layout, and Preview) that you use to work with your file.

**Mode pop-up menu**

A pop-up menu at the bottom of the document window from which you can choose a mode (Browse, Find, Layout, or Preview). This menu is available in all modes.

**Multi-key field**

A match field that contains more than one value, each on a separate line. A multi-key field can be used in one table involved in a relationship, to match several possible values in the match field of the other table. Sometimes called a *complex key field*.

**N****Navigation part**

A layout part that appears above a title header part (*top navigation part*) or below a title footer part (*bottom navigation part*), and can contain fields, objects, and other controls to help users navigate a layout or solution. Navigation parts do not print or preview.

**Network protocol**

A network protocol (for example, TCP/IP) is a set of rules that govern how computers exchange messages on a network.

**New Layout/Report assistant**

The New Layout/Report assistant guides you through creating a layout or report according to options you choose. In Layout mode, click **New Layout/Report** in the status toolbar, or choose **Layouts** menu > **New Layout/Report**.

## O

**Object**

On a FileMaker Pro layout, an object is a discrete entity or shape that you can select, move, modify, delete, or name. Lines, fields, buttons, popovers, panel controls, portals, imported graphics, blocks of text, tab controls, and web viewers are objects. Some objects, such as portals, have multiple components; for example, a portal row is a component of a portal.

**Object component**

An element of an object, such as a portal row of a portal or a slide panel of a slide control.

**ODBC**

An API that uses SQL statements to access data from, and exchange data with, many database management systems. FileMaker Pro uses ODBC drivers to share data (as a data source) and to interact with data from other applications (as a client application).

**1-away relationship**

A correspondence between database tables in which two tables are directly related to each other, with no other tables between them.

**One-to-many relationship**

A correspondence between data in database tables in which one record in the first table is related to more than one record in another table.

**One-to-one relationship**

A correspondence between data in database tables in which one record in the first table is related to one record in another table.

**Operands**

Components of a formula. For example, in the formula Quantity\*Price, Quantity and Price are the operands.

**Operators**

In calculations, symbols that indicate how to combine two or more expressions. These include the standard arithmetic operators (+, -, /, \*), logical operators that set up conditions that must be met to make a value true or false (AND, OR, XOR, and NOT), and find operators (<, =, @) that help you limit the records defined in a find request.

In the relationships graph, symbols that define the match criteria between one or more pairs of fields in two tables. These include: != (not equal), > (greater than), < (less than), = (equal), <= (less than or equal to), >= (greater than or equal to) and x (all rows, or Cartesian product).

## P

**Panel control**

Panel controls include the tab control and slide control. Individual panels of these controls are tab panels and slide panels, respectively.

**Parent script**

A script that defines script parameters and can call other scripts.

**Part label**

In Layout mode, the label that appears at the left or side of the bottom dividing line of each layout part. By dragging it up or down, you can use the part label to resize a part. You can also open the Part Definitions dialog box by double-clicking the label.

**Permitted host**

Servers, hosts, websites, services, and so on that are allowed to be connected to.

**PHP (PHP: Hypertext Preprocessor)**

An open-source programming language primarily used in server-side application software to create dynamic webpages. FileMaker Server lets you publish data from FileMaker Pro files on customized webpages created with PHP.

**Placeholder text**

A type of field label that displays within a field. Placeholder text can be a field name, other text (such as instructions), or the result of a calculation. In Layout mode, placeholder text is indicated by a Placeholder Text badge on a field.

**Plug-in**

Software that extends the capabilities of an application in a specific way.

**Popover**

A layout object that can contain other layout objects. Popovers include a content area (where objects are placed) and can include a title. You open a popover by clicking a popover button. Popovers reposition on the screen as needed to stay in view.

**Popover button**

A layout object that opens a popover.

**Port**

A pre-assigned number that indicates a “logical connection place” where a client (such as a web browser) can connect to a particular server application on a networked computer. Port numbers range from 0 to 65536. Port 80 is the default port for HTTP services such as FileMaker Pro web publishing, but you can use another port number if 80 is already in use by another server application.

**Portal**

For relational databases, a layout object in one table where you place one or more related fields to display in rows the data from one or more related records.

**Preview mode**

The FileMaker mode in which you see how layouts will look when they're printed.

(Use Browse mode to enter and edit your information; use Layout mode to design how your information is displayed. Use Find mode to find records that match search criteria; use Preview mode to display how your records will print.)

**Primary file**

In a FileMaker Pro Advanced runtime solution, the file that connects all of the auxiliary files and opens when you start the runtime application. From the Developer Utilities in FileMaker Pro Advanced, you can select a primary file for solutions that have more than one file.

**Privilege set**

A defined set of permissions that determines a level of access to a file. You can define as many privilege sets as you like for a file. There are three predefined privilege sets: Full Access, Data Entry Only, and Read-Only Access.

**Problem**

In the Script Workspace, an error, omission, or other inaccuracy found in a script.

## Q

### Query

Retrieving, manipulating, or modifying data from a data source by sending SQL statements. Also, requesting, and then receiving, data from a DBMS. You can also add, edit, format, sort, and perform calculations on your data using queries.

### Quick find

In Browse mode, searches records across multiple fields on a layout.

## R

### Read-Only Access privilege set

One of the three predefined privilege sets that appear in every file. The Read-Only Access privilege set allows read access to the records in a file, but not write or design access.

### Record

One set of fields in a database table. Each record contains data about a single activity, individual, subject, or transaction.

### Recover

If a file is damaged (for example, from an unexpected quit during a hard-drive update), you can attempt to recover the file. FileMaker Pro preserves as much data (the file's schema and structure and its tables, records, layouts, scripts, and field definitions) as possible.

### Recurring import

A method of setting up a file to automatically import read-only data from another file. With recurring import, imported data refreshes when you open the file, view the layout that contains the imported data for the first time during a FileMaker Pro session, or run a data update script.

### Recursive script

A script that calls itself.

### Related field

For relational databases, a field in one table that is related to a field in another table (or to a different field within the same table). If a relationship is defined between two tables (even through another table), data in fields in one table can be accessed from the other table.

### Related record

A record in the related table whose match field (according to the relationship used) contains a value that's equal to the value in the match field of another table.

### Related table

For relational databases, the table that contains the data you want to access and work with in the current table. For lookups, the table that contains the data to copy.

### Relational database

A group of one or more databases that, when used together, contain all the data you need. Each occurrence of data is stored in only one table at a time, but can be accessed in any table, either in the same file or from a related file. Data from another table or file is displayed in the current table without being copied, and the data changes whenever the values in the other table or file change.

## Relationship

Relationships provide access to data from one table to another. Relationships can join one record in one table to one record in another table, one record to many other records, or all records in one table to all records in another table, depending on the criteria you specify when you create the relationship in the relationships graph.

## Relationships graph

In the Relationships tab of the Manage Database dialog box, you can see the occurrences of tables both in the current file and from any external, related files. In this relationships graph, you join tables and change relationships between fields in different tables.

When you create a new table, a visual representation, or occurrence, of the table appears in the relationships graph. You can specify multiple occurrences (with unique names) of the same table in order to work with complex relationships in the graph.

## Repeating field

A field containing multiple, separate values.

## Report layout

A predefined layout type for setting up reports. You can create a layout with simple rows and columns of data (as in a list), or a complex report with grouped data (subsummary reports). The fields that you specify appear in columns across the screen or page in one line. Field names are in the header part and the footer part is blank.

## Report with grouped data

A subsummary report that you create using the Report layout type. Reports with grouped data can include totals and subtotals.

## Row

When a layout is viewed in Table View, a row corresponds to a record.

## Runtime solution

A solution that does not require FileMaker Pro or FileMaker Pro Advanced in order to be used. In FileMaker Pro Advanced, use the Developer Utilities to bind a primary file and any auxiliary files to produce a standalone runtime solution.

# S

## Schema

In database terminology, a schema is the organization of the tables, the fields in each table, and the relationships between fields and tables.

## Screen stencils

In Layout mode, nonprinting guides that help you design layouts for iPad, iPhone, iPod touch, or computers with different screen resolutions.

## Script

One or more instructions (script steps) that you define to automate repetitive or difficult tasks. You create and manage scripts using the Script Workspace. You run a script by clicking its button, choosing its menu command, calling it from another script or a plug-in, or running it at startup or when a file closes.

## Script Debugger

A FileMaker Pro Advanced tool for debugging FileMaker Pro scripts.

## Script step

A command that you include in a script.

**Script trigger**

A mechanism that causes a specified script to run when a particular event occurs.

**Search criteria**

In Find mode, the values and operators you specify to locate records. For example, if you type ABC Travel in the Vendor field, FileMaker Pro looks for and returns all records that have this name in the Vendor field.

**Security**

The protection that's placed on a file. Security includes various types of accounts to authenticate users, levels of privilege sets to determine what can be done with a file, and authorization of other files to create references to the current file (including its tables, layouts, scripts, and value lists). Security also includes extended privileges, which determine the data sharing options that are permitted by a privilege set.

**Segment**

A component of a button bar object. Each segment can be defined as a button or popover button.

**Selected tab**

In the Script Workspace, the tab that's in the foreground, in which you can create or edit a script.

**Self-join**

A relationship between fields in the same table. This creates another occurrence of the table in the relationships graph.

**Separator**

A line within a menu that separates or groups menu items. A separator can also be the character used to separate parts of a date, time, or number, or the semicolon (;) used to separate parameters in a function definition.

**Serial number**

A unique number entered by FileMaker Pro for each record. You can tell FileMaker Pro to automatically enter a serial number for each record by setting the Auto-Enter options in the Options for Field dialog box. You can also serialize records in Browse mode by choosing **Records** menu > **Replace Field Contents**.

**Shared file**

A file for which sharing has been enabled, which permits users to access the file over a network. FileMaker Pro, FileMaker Pro Advanced, and FileMaker Server each support one or more of the following ways to share files: FileMaker Network sharing, which permits multiple FileMaker Pro or FileMaker Go users to use a file simultaneously; publishing of files to web browser users via FileMaker WebDirect or Custom Web Publishing; and sharing of data with other applications via ODBC/JDBC.

**Shared ID**

In FileMaker Pro Advanced, a case-sensitive ID from 1 to 32 characters long that links encrypted files in a multfile solution.

**Shortcut**

Also known as keyboard shortcut. One or more keys that users can press to perform tasks.

**Shortcut menu**

Use to edit objects or data quickly by choosing commands from a shortcut, or context, menu. Commands vary depending on the mode you're using, the item the cursor is over, and whether an item is selected.

To display a shortcut menu, right-click (Windows) or Control-click (OS X) the item.



## Slide control

A layout object made up of one or more slide panels, allowing you to organize fields and other objects within each slide panel's borders.



## Slide panel


A component of a slide control. The slide panel is the area displayed when a dot in a slide control is selected. You can place objects such as lines, fields, buttons, portals, imported graphics, blocks of text, tab controls, slide controls, and web viewers on slide panels.

## Slider

In the status toolbar, the navigation control for quickly moving to a record in your file based on its location in the file. In Browse mode, moving the slider changes the current record. In Find mode, moving the slider changes the current find request. In Layout mode, moving the slider changes the current layout. In Preview mode, moving the slider changes the current page.

## Sliding objects

Objects that move together to close gaps left by entries in adjacent fields.

Set sliding in Layout mode, in the Sliding & Visibility area of the Position  tab in the Inspector.

## Snapshot link

A found set of records that is saved in the FileMaker Pro Snapshot Link (FMPSL) format, with the filename extension .fmpl. A snapshot link captures and preserves the found set as it was when you performed the find request. You can also email an FMPSL file to another person.

## Solution

A file or a set of files containing database tables, layouts, scripts, and associated data. A solution solves one or more specific problems, such as tracking customers or invoices.

## Sort order

The sequence for rearranging records. Records are sorted by the first field in the sort order list, then the second, and so on. Values within each field are sorted by the order specified (ascending, descending, or custom).

## Source file

The file from which you bring data during importing or exporting, or the file from which you add a table to the relationships graph.

## Source table


The table upon which one or more table occurrences in the relationships graph are based. The source table is the table defined in the Tables tab of the Manage Database dialog box.

## SQL

A structured programming query language that controls and interacts with a DBMS.

## Stacking order

The order in which objects overlap on a layout. In Layout mode, you can change this order by cutting and pasting objects or by clicking Arrange buttons in the Arrange & Align area of the

Position  tab in the Inspector.

## Standard form layout

The default layout, with all fields arranged on separate lines in the order they were defined. The body part is only as tall as it needs to be to include all the fields in the file. This layout includes header and footer parts. (In previous versions of FileMaker Pro, a Standard Form layout was a predefined layout type.)

## Starter Solution

A professionally designed FileMaker Pro file customized for managing common business or personal tasks on iPad, iPhone, desktop, or the web.

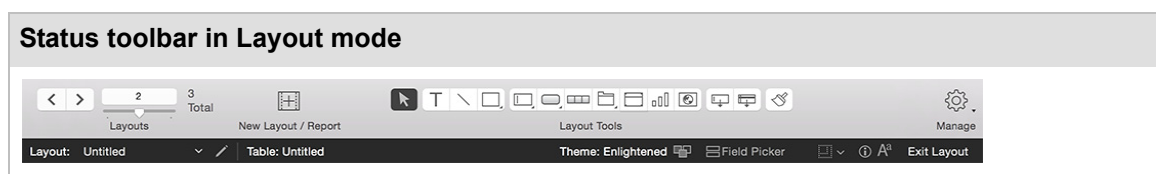
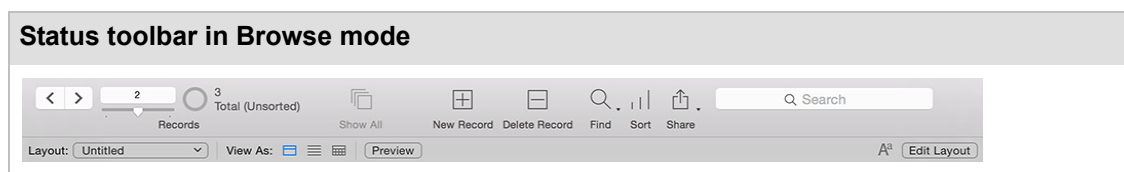
## Startup script


A script that automatically runs when a file is opened. You can script such things as setting system formats to the user's formats or setting a file to be shared in a startup script.

You specify a startup script in the File Options dialog box.

## Status toolbar

The area across the top of the document window that displays navigation controls, customizable buttons, and a layout bar for working with layouts. In Layout mode, it includes layout tools.



If you don't see the status toolbar, click the status toolbar control  at the bottom of the document window.

## Structure

In FileMaker Pro, the organization of file elements such as scripts, layouts, value lists, and privileges. You interact with a file's schema through its structure.

## Style

A collection of attributes, such as font, color, line style, and text alignment, that determines the appearance of a layout.

## Submenu

A menu that extends from another menu item.

## Sub-script

A script that is called from another script.

### Subsummary parts

Use summary parts to view and display information from one or more records. You place a summary field in a summary part to display a summary of information for each group of records sorted on the break field. You can add one or more subsummaries above (leading) or below (trailing) the body.

### Subsummary value

Aggregate values for different categories of data within a field. For example, a subsummary value can be the total of employees for each department.

### Summary field

A field that contains the result of a summary calculation of values across a group of records.

### Supplemental field

A FileMaker calculation field or summary field that you can append to ODBC tables in order to do calculations on the external data while working in FileMaker. The calculations are not stored and you are not changing the schema of the ODBC table.

### System formats

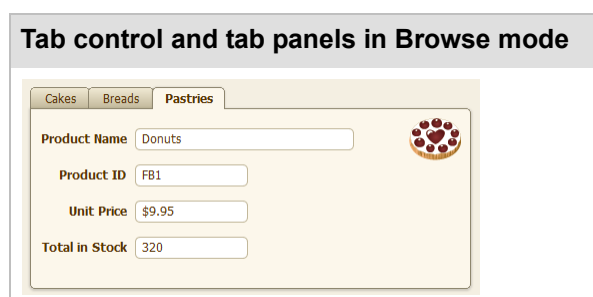
Settings that determine how your operating system displays and sorts dates, times, numbers, and currency.

If the system formats are different on your computer from those on the computer on which the file was created, the first time you open the file, FileMaker Pro asks if you want to use the system's settings or the file's settings. When you format fields, you can use the Inspector to have data displayed according to the current system formats.

## T

### Tab control

A layout object made up of one or more tab panels that allows you to organize fields and other objects within each tab panel's borders.



### Tab order

The order in which you move from field to field in a record. In Layout mode, you can define a custom tab order and include buttons, panel controls, and web viewers in the tab order.

### Tab panel

A component of a tab control. The tab panel is the area displayed when a tab in a tab control is selected. You can place objects such as lines, fields, buttons, portals, imported graphics, blocks of text, tab controls, slide controls, and web viewers on tab panels.

## Table

A collection of data pertaining to a subject, such as customers or stock prices. A database contains one or more tables, which consist of fields and records. When you create a new table, a visual representation, or occurrence, of the table appears in the relationships graph. You can specify multiple table occurrences (with unique names) of the same table in order to work with complex relationships in the graph.

## Table View

Displays multiple records in a tabular format like a spreadsheet. Each record appears in a row, and each field appears in a column. To select this view, click **Table View** in the layout bar.

In Browse mode, you can use Table View to create, modify, and delete fields; choose field types; add, delete, and sort records; or create a chart or dynamic report.

## Target file

The file into which you bring data during importing or exporting.

## TCP/IP (Transmission Control Protocol/Internet Protocol)


The basic communication protocol that is the foundation of the Internet.

## Template

A predefined website that you can select in the Web Viewer Setup dialog box to help you create a web viewer quickly.

## Text baseline

In Layout mode, the guideline that appears at the base of the text in a field or text block. Text baselines can be solid, dotted, or dashed. If you want text baselines to also appear in Browse

and Find modes, select **Text baselines** in the Appearance  tab in the Inspector.

## Text expression

Any expression that returns a text result. For example, a text expression can be a constant ("London"), a field reference (Status), or a calculated value (Rightwords(Lastname;1)).

## Timestamp

A field type combining date and time that is compatible with the ODBC requirement for the SQL format [yyyy.mm.dd hh:mm:ss.sss].

## Tooltip

A small box that displays text when a user moves the cursor over a layout object. Tooltips display in Browse, Find, and Layout modes.

# U

## Unicode

A worldwide standard that, in one code page, provides a unique number for every character in human languages, no matter what the platform, software program, or operating system.

## Unit of measure

In Browse and Layout modes, you can set the unit of measure to points, inches, or centimeters.

## Unstored calculation

A calculation field with a result that is only calculated when the value is needed, for example, to browse or print. In most cases, FileMaker Pro makes a field stored when you define it, but you can change the storage type to unstored.

## URL (Uniform Resource Locator)

A web address, which consists of a protocol, a host name, and optionally a port, a directory, and a filename. For example, `http://www.filemaker.com/`, `ftp://12.34.56.78:80/myfiles/`, or `fmp://mywebsite.com/sample.fmp12`.

## V

### Value list

To save time and ensure accuracy during data entry, define frequently used text, number, date, or time values as a value list. When you enter data, you can choose from the list of defined values.

You can format value lists to display in a drop-down list or pop-up menu, or as checkboxes or option (radio) buttons. The values in a value list can be user-defined or based on the values in a field in the same file or in a different file. You can also define relationships for use with value lists, to access and display particular related values. Another option is to use a value list from another file.

### Variable

In a calculation, a symbol or name that represents a value. Use the Set Variable script step to specify the name, value, and repetition of the variable. Names prefixed by \$ are local variables available only within the current script. Prefix the name with \$\$ to make the variable available throughout the current file (global). Local and global variables can have the same name but they are treated as different variables.

### View

An arrangement of your data primarily useful for onscreen manipulation. In Browse mode, Find mode, or Preview mode, Form View displays individual records, List View displays records in a list, and Table View displays records in a spreadsheet-like table format.

## W, X, Y, Z

### Web address

The calculated expression that you enter in the Web Viewer Setup dialog box. A web address is not equivalent to a URL that a web user could enter in a web browser.

### Web browser

An application that you can use to view webpages or websites on the World Wide Web or an intranet. Browsers download the webpages to the viewer's computer.

### Webpage

An HTML document displayed on the Internet or on an intranet.

### Web server

A computer that is connected to the Internet or an intranet and has a web server application installed on it. Web server applications deliver webpages and associated files to web browsers.

### Website

One or more webpages connected by links and displayed on the Internet or an intranet.

### Web user

Someone using a web browser to access a FileMaker Pro file published on the World Wide Web or an intranet.

**Web viewer**

A layout object that allows you to display information from websites based on data in your file.

**World Wide Web**

An interlinked collection of webpages residing on web servers, and other documents, menus, and files, which are available via URLs.

**X-axis data**

In a column, stacked column, line, and area chart, the data series you are comparing (for example, company name).

In a bar or stacked bar chart, the data series you are measuring (for example, annual sales).

**XSLT (Extensible Stylesheet Language Transformations)**

XSLT (XSL Transformations) is a subset of XSL (Extensible Stylesheet Language) that is used to transform, or change, the structure of an XML document into a different document format. For example, you can use an XSLT style sheet to transform an XML document into an HTML or TXT document.

**XML (Extensible Markup Language)**

Instead of being a rigid file format, XML is a language for defining agreed-upon formats that groups can use for exchanging data. Many organizations and businesses are using XML to transfer product information, transactions, inventory, and other business data.

FileMaker Pro can export XML data that can then be used, for example, by spreadsheet applications, data charting applications, and enterprise SQL databases. FileMaker Pro can also import XML data.

**Y-axis data**

In a column, stacked column, line, and area chart, the data series you are measuring (for example, annual sales).

In a bar or stacked bar chart, the data series you are comparing (for example, company name).