Functions

AddBullet

`AddBullet`

AddBullet["bullet",nb] adds the bullet "bullet" as the CellDingbat for the selected cell in the notebook nb. AddBullet["b
"bullet" as the CellDingbat for the selected cell in the current InputNotebook[].

Attributes for `AddBullet`

```
{Locked, Protected, ReadProtected}
```

AddCellFrameLabelToCell

`AddCellFrameLabelToCell`

AddCellFrameLabelToCell[nb,data,place] adds a cell frame label to the selected cell. The location of the CellFrameLab
which can be Left, Right, Top, or Bottom. If the given cell already has a cell frame label in the specified place it will b
provided. "data" should either be a String or a Cell.

Attributes for `AddCellFrameLabelToCell`

```
{Locked, Protected, ReadProtected}
```

AddCellStylesToDiaryEntryPalette

`AddCellStylesToDiaryEntryPalette`

AddCellStylesToDiaryEntryPalette["CellStyle"] adds the CellStyle to the Diary Entry palette.

Attributes for `AddCellStylesToDiaryEntryPalette`

```
{Locked, Protected, ReadProtected}
```

AddCellStylesToEssayPalette

`AddCellStylesToEssayPalette`

AddCellStylesToEssayPalette["CellStyle"] adds the CellStyle to the Essay palette.

Attributes for `AddCellStylesToEssayPalette`

```
{Locked, Protected, ReadProtected}
```

AddCellTag

`AddCellTag`

AddCellTag[nb,tag] adds the tag "tag" to the CellTags of the currently selected cell in the notebook nb if that cell doesn'
its CellTags. A related function is TagCell.

Attributes for `AddCellTag`

```
{Locked, Protected, ReadProtected}
```

AddCellTaggingRule

`AddCellTaggingRule`

AddCellTaggingRule[nb,tag] adds "tag" to the TaggingRules for the selected cell in the notebook object nb if it is not al
TaggingRules.

Attributes for `AddCellTaggingRule`

```
{HoldRest, Locked, Protected, ReadProtected}
```

AddCurrentPackageToPluginsDirectory

`AddCurrentPackageToPluginsDirectory`

AddCurrentPackageToPluginsDirectory[] adds a copy of the current package to the Plugins directory. The original pack
original place in PackagesDirectory[].

Attributes for `AddCurrentPackageToPluginsDirectory`

```
{Locked, Protected, ReadProtected}
```

AddDashboardElements

`AddDashboardElements`

AddDashboardElements[{"elements"...}] adds the elements to the Dashboard.

Attributes for `AddDashboardElements`

```
{Locked, Protected, ReadProtected}
```

AddDatabaseFields

`AddDatabaseFields`

AddDatabaseFields[name, {fieldNames...}] adds the new fields to the already existing database. The database must be lo
this function. The new fields are populated according to the option DefaultDatabaseFieldValues. RestoreDatabase has
AddDatabaseFields has been executed. However, the option BackupFirst has the default value BackupFirst→True anc
recovered from its backup.

Default options for `AddDatabaseFields`

```
{DefaultDatabaseFieldValues → GUID, BackupFirst → True}
```

Attributes for `AddDatabaseFields`

```
{HoldFirst, Protected, ReadProtected}
```

AddDatabaseRecords

`AddDatabaseRecords`

AddDatabaseRecords[name,{records...}] adds the records to the database given by name. These records are listed in Nev
database is reloaded, whereupon the records are added to the database file. Prior to that the new records are stored in a
DatabaseDirectory[name].

Attributes for `AddDatabaseRecords`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

AddDatabaseRecordsToClosedDatabase

AddDatabaseRecordsToClosedDatabase

AddDatabaseRecordsToClosedDatabase AddDatabaseRecordsToClosedDatabase[name,"directory", {records...}] adds t even if the database has not been loaded. The directory given should be the path to the database. AddDatabaseRecord not check the FieldTypes of the records that are added to the closed database. Hence, other precautions should be take records being added conform to the FieldTypes of the given database.

Attributes for `AddDatabaseRecordsToClosedDatabase`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

AddDiaryKeywords

AddDiaryKeywords

AddDiaryKeywords[{"keyword1","keyword2",...}] adds the keywords to the list of those that the package uses to identi directory $CurrentDiaryNotebookDirectory. AddDiaryKeywords[All] allows all possible keywords.

Attributes for `AddDiaryKeywords`

```
{Locked, Protected, ReadProtected}
```

AddDueDate

AddDueDate

AddDueDate[nb date] adds a DueDate tag to the selected ToDo cell in the notebook nb.

Attributes for `AddDueDate`

```
{Locked, Protected, ReadProtected}
```

AddEssayBodyCell

AddEssayBodyCell

AddEssayBodyCell[nb,guid,style]

Attributes for `AddEssayBodyCell`

```
{Locked, Protected, ReadProtected}
```

AddEssayNotesCell

AddEssayNotesCell

AddEssayNotesCell[nb,guid]

Attributes for `AddEssayNotesCell`

```
{Locked, Protected, ReadProtected}
```

AddExtraButtonsToPalette

AddExtraButtonsToPalette

AddExtraButtonsToPalette[$PaletteExtraButtonsParameter,otherButtonInformation1,otherButtonInformation2,...] assig
the palette if it allows for them. Each button's information should be of the form: {_String,_Function|None,{___?Opt
This is the same form as is returned by the function SavedButtonInformation.

Attributes for `AddExtraButtonsToPalette`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

AddFileTo$FileSetsDialog

AddFileTo$FileSetsDialog

AddFileTo$FileSetsDialog[] opens a dialog that allows you to name FileSets and add files to them and hence to $FileSe

Attributes for `AddFileTo$FileSetsDialog`

```
{Locked, Protected, ReadProtected}
```

AddFormattingBackgroundColors

AddFormattingBackgroundColors

AddFormattingBackgroundColors[{{"name",color},...}] adds the colors to the formatting palette. "name" should be stri
color directives (RGBColor,GrayLevel,Hue, or CMYKColor). To interactively create a color use the function PasteC

Attributes for `AddFormattingBackgroundColors`

```
{Locked, Protected, ReadProtected}
```

AddFormattingTextColors

AddFormattingTextColors

AddFormattingTextColors[{{"name",color},...}] adds the colors to the formatting palette. Each "name" should be a stri
color directive (RGBColor,GrayLevel,Hue, or CMYKColor). To interactively create a color use the function PasteCo

Attributes for `AddFormattingTextColors`

```
{Locked, Protected, ReadProtected}
```

AddGUIDTagToCellTaggingRules

AddGUIDTagToCellTaggingRules

AddGUIDTagToCellTaggingRules[nb] adds a GUIDTag to the TaggingRules of the selected cell in the notebook nb if i
The returned value is the GUIDTag that has been added (or, if the cell already has one, it is the value of that tag).

Attributes for `AddGUIDTagToCellTaggingRules`

```
{Locked, Protected, ReadProtected}
```

AddGUIDToNotebook

AddGUIDToNotebook

AddGUIDToNotebook[nb] adds a GUID to the TaggingRules of the notebook object nb if it doesn't already have one.

Attributes for `AddGUIDToNotebook`

```
{Locked, Protected, ReadProtected}
```

AddGUIDToNotebookAndSave

AddGUIDToNotebookAndSave

AddGUIDToNotebookAndSave is an option to NotebookDiscovery that specifies whether notebooks found by Noteboo
tagged with a GUID (to make it able to be located if its location is changed), resaved, and closed. Its default value is F
True the process may take a somewhat longer than with its value at False.

Attributes for `AddGUIDToNotebookAndSave`

```
{Locked, Protected, ReadProtected}
```

AddHeading

AddHeading

AddHeading[heading,style] adds the heading "heading" to the list of headings of style "style".

Attributes for `AddHeading`

```
{Locked, Protected, ReadProtected}
```

AdditionalToolsPalette

AdditionalToolsPalette

AdditionalToolsPalette[] opens the "Additional Tools" Palette. Additional buttons can be appended to this palette by ass
to $AdditionalToolsPaletteExtraButtons and executing AdditionalToolsPalette[Sequence@@$AdditionalToolsPalette

Attributes for `AdditionalToolsPalette`

```
{Locked, Protected, ReadProtected}
```

AddMetaDataCell

AddMetaDataCell

AddMetaDataCell[nb] adds at MetaDataCell to the notebook object nb. Its return value is the GUID of the metadata cell
one metatdata cell with a given GUID is allowed in a notebook.

Attributes for AddMetaDataCell

```
{Locked, Protected, ReadProtected}
```

AddMetaDataToMetaDataCell

AddMetaDataToMetaDataCell

AddMetaDataToMetaDataCell[nb,guid,{metadata...}] adds the metadata to the MetaDataCell with the GUID guid. Only not already included in the MetaDataCell are added. I.e., the metadata in the MetaDataCell are not repeated. AddMetaDataToMetaDataCell[nb,guid,metaDatum] adds the single item of metaData, metaDatum, which is not a list

Attributes for AddMetaDataToMetaDataCell

```
{Locked, Protected, ReadProtected}
```

AddNotebookTaggingRule

AddNotebookTaggingRule

AddNotebookTaggingRule[nb,tag] adds "tag" to the TaggingRules for the notebook object nb if it is not already amongs

Attributes for AddNotebookTaggingRule

```
{HoldRest, Locked, Protected, ReadProtected}
```

AddNotebookToNotebooksMenu

AddNotebookToNotebooksMenu

AddNotebookToNotebooksMenu[nb] adds the file corresponding to the open notebook object nb to the File▹Open Rece

Attributes for AddNotebookToNotebooksMenu

```
{Locked, Protected, ReadProtected}
```

AddOpenNotebooksTo$FileSets

AddOpenNotebooksTo$FileSets

AddOpenNotebooksTo$FileSets["name"] adds all currently open notebooks to the FileSet with the given name. Only th
neither palettes nor Dialogs and which have been saved will be added.

Attributes for `AddOpenNotebooksTo$FileSets`

```
{Locked, Protected, ReadProtected}
```

AddPackageFunctionCategory

`AddPackageFunctionCategory`

AddPackageFunctionCategory["category"] adds a function category to the $CurrentPackageNotebook.

Attributes for `AddPackageFunctionCategory`

```
{Locked, Protected, ReadProtected}
```

AddRSSFeedDialog

`AddRSSFeedDialog`

AddRSSFeedDialog[] opens the Add to RSSFeed Dialog so that you can add an RSS Feed to the RSSFeedsPalette.

Attributes for `AddRSSFeedDialog`

```
{Locked, Protected, ReadProtected}
```

AddSaveBackupToolbarCell

`AddSaveBackupToolbarCell`

AddSaveBackupToolbarCell[nb] adds the SaveBackupToolbarCell to the top of the notebook nb and removes previous
are in the notebook. AddSaveBackupToolbarCell[] performs this function for the notebook that it is evaluated in.

Attributes for `AddSaveBackupToolbarCell`

```
{Locked, Protected, ReadProtected}
```

AddTimeEstimate

AddTimeEstimate

AddTimeEstimate[nb,estimate] adds a TimeEstimate tag to the specified ToDo cell in the notebook nb.

Attributes for `AddTimeEstimate`

```
{Locked, Protected, ReadProtected}
```

AddTimeTaken

AddTimeTaken

AddTimeTaken[nb,estimate] adds a TimeTaken tag to the specified Done cell in the notebook nb.

Attributes for `AddTimeTaken`

```
{Locked, Protected, ReadProtected}
```

AddToHideTag

AddToHideTag

AddToHideTag[nb] adds the tag "ToHide" to the CellTags of the currently selected cell in the notebook nb if that cell d
amongst its CellTags.

Attributes for `AddToHideTag`

```
{Locked, Protected, ReadProtected}
```

AddTo$FavoriteDiaries

AddTo$FavoriteDiaries

AddTo$FavoriteDiaries[] adds the current Diary to the list of favorite Diaries.

Attributes for `AddTo$FavoriteDiaries`

```
{Locked, Protected, ReadProtected}
```

AddTo$FavoriteNotebooks

`AddTo$FavoriteNotebooks`

AddTo$FavoriteNotebooks[] adds the current Diary to the list of favorite Notebooks.

Attributes for `AddTo$FavoriteNotebooks`

```
{Locked, Protected, ReadProtected}
```

AddTo$FileSets

`AddTo$FileSets`

AddTo$FileSets["name",file] adds the given file to the FileSet with the given name. AddTo$FileSets[name,{file1,file2,.
FileSet with the given name.

Attributes for `AddTo$FileSets`

```
{Locked, Protected, ReadProtected}
```

AddTo$RecentDiaries

`AddTo$RecentDiaries`

AddTo$RecentDiaries[] adds the current Diary to the list of recent Diaries.

Attributes for `AddTo$RecentDiaries`

```
{Locked, Protected, ReadProtected}
```

AddTo$RecentNotebooks

`AddTo$RecentNotebooks`

AddTo$RecentNotebooks[] adds the input notebook to the list of recent Notebooks. AddTo$RecentNotebooks[nb] adds of recent Notebooks.

Attributes for `AddTo$RecentNotebooks`

```
{Locked, Protected, ReadProtected}
```

AddTo$RSSFeeds

AddTo$RSSFeeds

AddTo$RSSFeeds["name","url"]

Attributes for `AddTo$RSSFeeds`

```
{Locked, Protected, ReadProtected}
```

AddTo$TaggingList

AddTo$TaggingList

AddTo$TaggingList[{tags...},tf] AddTo$TaggingList appends to $TaggingList those tags in the list of those in its first
argument, tf, determines whether or not the TaggingPalette should be refreshed. Its value should be either True or Fal
AddTo$TaggingList[{tags...}] is equivalent to AddTo$TaggingList[{tags...},True].

Attributes for `AddTo$TaggingList`

```
{Locked, Protected, ReadProtected}
```

AggregateToDos

AggregateToDos

AggregateToDos[] collects all ToDos in the current Diary notebook and places them at the beginning of the notebook. T
sorted by the time that they were marked as a ToDo and are under a Section heading that reads "To Dos." You may w
BackupNotebook[CurrentDiary] to back up the current Diary before aggregating its ToDos. This is generally done Au
setting of the option BackupFirst→True. "AggregateToDos" is also used as the Cell tag of the Section cell that preced

Default options for `AggregateToDos`

```
{BackupFirst → True}
```

Attributes for `AggregateToDos`

```
{Protected, ReadProtected}
```

AlignCellText

AlignCellText

AlignCellText[x, nb] algins the contents of the current cell in the notebook nb. For example AlignCellText[Center, nb] c
possible values of x see the Mathematica documentation for the cell option TextAlignment. AlignCellText[Default,nb
to the default value for a cell of its type. AlignCellText[x] and AlignCellText[Default] have the described effect on th
InputNotebook[].

Attributes for `AlignCellText`

```
{Locked, Protected, ReadProtected}
```

AllOpenDiaries

AllOpenDiaries

AllOpenDiaries[] gives a list of all currently open notebooks that are Diaries.

Attributes for `AllOpenDiaries`

```
{Locked, Protected, ReadProtected}
```

AllOpenNotebooksPalette

AllOpenNotebooksPalette

AllOpenNotebooksPalette[] pops up a palette listing all of the currently open notebooks in the Mathematica session (exc
AllOpenNotebooksPalette itself). Clicking on a notebook's button in this palette makes that notebook the currently sel

Attributes for `AllOpenNotebooksPalette`

```
{Locked, Protected, ReadProtected}
```

AllOrganizations

`AllOrganizations`

AllOrganizations[] gives the list of Organizations. This is in the form of a list of triplets. For each triplet the first item i
(a String with no WhiteSpace), the second one is a Directory that is associated with the Organization or None if there
the third is a set of rules associated with the Organization.

Attributes for `AllOrganizations`

```
{Locked, Protected, ReadProtected}
```

AllPalettesPalette

`AllPalettesPalette`

AllPalettesPalette[] opens the AllPalettesPalette.

Attributes for `AllPalettesPalette`

```
{Locked, Protected, ReadProtected}
```

AnalyticsPalette

`AnalyticsPalette`

AnalyticsPalette[] opens the Analytics Palette.

Attributes for `AnalyticsPalette`

```
{Locked, Protected, ReadProtected}
```

AppendTo$Path

`AppendTo$Path`

AppendTo$Path[dir] appends the directory dir to the list $Path if is not already in that list. dir must be a string.

Attributes for `AppendTo$Path`

```
{Locked, Protected, ReadProtected}
```

ApplyDiaryTemplateToDiary

`ApplyDiaryTemplateToDiary`

ApplyDiaryTemplateToDiary["name"] applies the template with name "name." to the current Diary. To find out the nam
execute the function DiaryTemplates[]. If $DeleteDefaultCodeCellBeforeApplyDiaryTemplateToDiary is set to True,
code cells in the current Diary notebook are replaced with those from the template. All default code cells are then eva
ApplyDiaryTemplateToDiary[Default] returns the default code cells to the default.

Attributes for `ApplyDiaryTemplateToDiary`

```
{Locked, Protected, ReadProtected}
```

ArchiveDate

`ArchiveDate`

ArchiveDate is a function head indicating a date tag for an Archive.

Attributes for `ArchiveDate`

```
{Locked, Protected, ReadProtected}
```

ArchiveDiary

`ArchiveDiary`

ArchiveDiary[] archives the current Diary notebook. The ToDos in the Diary are transferred to the current Diary, and th
removed and placed in the archive.

Attributes for `ArchiveDiary`

```
{Locked, Protected, ReadProtected}
```

ArchiveDiaryDialog

`ArchiveDiaryDialog`

ArchiveDiaryDialog[] opens a dialog that asks whether the user wants the current Diary to be archived.

Attributes for `ArchiveDiaryDialog`

```
{Locked, Protected, ReadProtected}
```

Archives

`Archives`

Archives[] gives a list of the archives of the current Diary. Archives[Notebook] opens up a notebook with information o
hyperlinks to them.

Attributes for `Archives`

```
{Locked, Protected, ReadProtected}
```

ArchivingButtonData

`ArchivingButtonData`

ArchivingButtonData[] gives a list of button information for use in AssignButtonsToCustomPalette.

Attributes for `ArchivingButtonData`

```
{Locked, Protected, ReadProtected}
```

AssignButtonsToCustomPalette

`AssignButtonsToCustomPalette`

AssignButtonsToCustomPalette[i,buttonInformationList] assigns to the ith custom Palette the buttons defined in the butt
buttonInformationList should be of the form {{String,Function}...} where the String gives the button its name and the
of the button when clicked upon.

Attributes for `AssignButtonsToCustomPalette`

```
{Locked, Protected, ReadProtected}
```

AssignToOrganization

`AssignToOrganization`

AssignToOrganization[nb,"organization"] assigns the notebook object nb to the organization "organization". When this notebook's OrganizationFlow is set to None. To set its OrganizationFlow to something else, use the function UpdateNotebookOrganizationFlow.

Attributes for `AssignToOrganization`

```
{Locked, Protected, ReadProtected}
```

BackupAndEncryptionButtonData

`BackupAndEncryptionButtonData`

BackupAndEncryptionButtonData[] gives a list of button information for use in AssignButtonsToCustomPalette.

Attributes for `BackupAndEncryptionButtonData`

```
{Locked, Protected, ReadProtected}
```

BackupDatabase

`BackupDatabase`

BackupDatabase[name] backs up the given database. The database must be currently loaded to do this backup.

Attributes for `BackupDatabase`

```
{Locked, Protected, ReadProtected}
```

BackupDirectory

`BackupDirectory`

BackupDirectory[dir] backs up the directory dir.

Attributes for `BackupDirectory`

```
{Locked, Protected, ReadProtected}
```

BackupDirectoryNameQ

`BackupDirectoryNameQ`

BackupDirectoryNameQ[dir] determines whether the form of the directory name "dir" is such that it could have been ge
BackupDirectory. This form is any directory with a name that is of the form string<>"BU"<>absoluteDateString. Bac
BUString] uses the user-supplied string BUString instead of "BU".

Attributes for `BackupDirectoryNameQ`

```
{Locked, Protected, ReadProtected}
```

BackupFileNameQ

`BackupFileNameQ`

BackupFileNameQ[file] determines whether the form of the file name "file" is such that it could have been generated by
form is any file with a name that is of the form string<>"BU"<>absoluteDateString<>".nb". BackupFileNameQ[file,
supplied string BUString instead of "BU".

Attributes for `BackupFileNameQ`

```
{Locked, Protected, ReadProtected}
```

BackupFirst

`BackupFirst`

BackupFirst is an option to AggregateToDos that determines whether to backup the $CurrentDiaryNotebook prior to ag
BackupFirst is also an option to DeleteDatabase, AddDatabaseFields, and DeleteDatabaseFields that determines whet
prior to modifying it according to these functions.

Attributes for `BackupFirst`

```
{Locked, Protected, ReadProtected}
```

BackupNotebook

BackupNotebook

BackupNotebook[nb] backs up the notebook given by the notebook object nb. The backed up notebook is saved to the sa
name of the backed up notebook is that of nb with "BU" and a time tag appended to it. The time tag is IntegerPart[Ab
tag the time in GMT. The form BackupNotebook[nb,BUString] uses BUString instead of "BU" BackupNotebook[nb,
subDirectory] backs up nb to the subDirectory of the directory of nb. subDirectory should be give as a list of strings. I
backup is saved to nb's directory. If, for example, subDirectory={"dir"} then the backup is saved to the directory "dir
"dir" doesn't exist then an error message is generated and $Failed is returned. BackupNotebook[] backs up the notebo

Attributes for `BackupNotebook`

```
{Locked, Protected, ReadProtected}
```

BitMapImage

BitMapImage

BitMapImage is an option for PlaceImage that determines whether the image is converted to a Bitmap. Its default value

Attributes for `BitMapImage`

```
{Locked, Protected, ReadProtected}
```

BlogNames

BlogNames

BlogNames[] gives the names of the blogs associated with the current Diary.

Attributes for `BlogNames`

```
{Locked, Protected, ReadProtected}
```

BlogPalette

BlogPalette

BlogPalette[] opens the blog palette.

Attributes for `BlogPalette`

```
{Locked, Protected, ReadProtected}
```

BlogState

`BlogState`

BlogState is an Option to DiaryToSimpleBlog that specifies whether the blog should be created with all the Diary entrie
Automatic open all Diary entries. The value Closed has only the most recent entry open with all others as hyperlinks.
entries as hyperlinks.

Attributes for `BlogState`

```
{Locked, Protected, ReadProtected}
```

BlogTemplates

`BlogTemplates`

BlogTemplates[] gives a list of the names of the blog templates in $BlogTemplatesDirectory.

Attributes for `BlogTemplates`

```
{Locked, Protected, ReadProtected}
```

BypassDiaryChecksOnSaveDiary

`BypassDiaryChecksOnSaveDiary`

BypassDiaryChecksOnSaveDiary[] sets $DateTagDiaryOnSaveDiary, $ContractDiaryNotebookOnSaveDiary, and $Loc
False for the $CurrentDiaryNotebook. It should generally be used in a DefaultCodeCell of the given Diary. Each of
$DateTagDiaryOnSaveDiary, $ContractDiaryNotebookOnSaveDiary, and $LockCellsOnSaveDiary are reset to True
but before the Default code cells of the Diary are executed. It is generally not advised to use BypassDiaryChecksOnS
diary is saved using SaveDiary[] (or the appropriate palette or toolbar button, a number of things are processed in the
provide information to the Worklife™ FrameWork's functions.

Attributes for `BypassDiaryChecksOnSaveDiary`

```
{Locked, Protected, ReadProtected}
```

CalendarDate

CalendarDate

CalendarDate[date] gives a string with the textual form of the calendar date. The form of the output is determined by the CalendarDate: DateOrder, TimeForm, IncludeDayOfTheWeek, IncludeTime, IncludeSeconds, and OriginatingTimeZ setting for OriginatingTimeZone is OriginatingTimeZone→0. I.e., the date is assumed to be given in Coordinated Un Mean Time"). CalendarDate[] gives the current Calendar Date in Coordinated Universal Time. CalendarDate[Origina gives the current Calendar Date in the time zone timeZone.

Default options for CalendarDate

```
{DateOrder :→ $DateOrder, TimeForm → 12, IncludeDayOfTheWeek → True,
 IncludeTime → False, IncludeSeconds → True, OriginatingTimeZone → 0}
```

Attributes for CalendarDate

```
{Protected, ReadProtected}
```

CellCreationHistory

CellCreationHistory

CellCreationHistory[nb] gives a sorted list of the dates that Cells in the notebook nb were marked as created. Dates are g your system settings.

Attributes for CellCreationHistory

```
{Locked, Protected, ReadProtected}
```

CellCreationHistoryPointStyle

CellCreationHistoryPointStyle

CellCreationHistoryPointStyle is an option to NotebookOpeningHistoryGraphicCell giving the style directives for the C in the graphic.

Attributes for CellCreationHistoryPointStyle

```
{Locked, Protected, ReadProtected}
```

CellCreationHistoryReport

CellCreationHistoryReport

CellCreationHistoryReport[nb] gives a list of properties of the history of the creation of Cells in the notebook nb. Dates
Time of your system settings.

Attributes for CellCreationHistoryReport

```
{Locked, Protected, ReadProtected}
```

CellTaggingRules

CellTaggingRules

CellTaggingRules[nb] gives the list of the TaggingRules for the selected cell in the notebook object nb (corresponding t
Cell's TaggingRules option).

Attributes for CellTaggingRules

```
{Locked, Protected, ReadProtected}
```

CenterGraphics

CenterGraphics

CenterGraphics[nb] centers a graphic in the selected cell.

Default options for CenterGraphics

```
{AutoScroll → True}
```

Attributes for CenterGraphics

```
{Protected, ReadProtected}
```

ChangeDatabaseFieldNames

ChangeDatabaseFieldNames

ChangeDatabaseFieldNames[name,fieldNames] changes the value of DatabaseFieldNames[name] to fieldNames.

Attributes for `ChangeDatabaseFieldNames`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

ChangeDatabaseFieldTypes

`ChangeDatabaseFieldTypes`

ChangeDatabaseFieldTypes[name,{fieldTypes...}]changes the field types for the database with the given name. {fieldTy
patterns (representing the field types) that is Length[DatabaseFieldNames[name]] long.

ChangeDatabaseFieldTypes[name,Automatic] makes all field types equal to the arbitrary pattern _.

ChangeDatabaseFieldTypes[name,String] makes all field types equal to the string pattern _String.

More generally,

ChangeDatabaseFieldTypes[name,head] makes all field types equal to the arbitrary pattern _head for the given head.

Attributes for `ChangeDatabaseFieldTypes`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

ChangePackageFunctionName

`ChangePackageFunctionName`

ChangePackageFunctionName[originalFunctionName,newFunctionName] changes the exported function with the name
into one with newFunctionName.

Attributes for `ChangePackageFunctionName`

```
{Locked, Protected, ReadProtected}
```

CheckDiaryNotebookFileAndDirectoryStatus

`CheckDiaryNotebookFileAndDirectoryStatus`

CheckDiaryNotebookFileAndDirectoryStatus[] is True if both $CurrentDiaryNotebookDirectory and $CurrentDiaryNot

Attributes for `CheckDiaryNotebookFileAndDirectoryStatus`

```
{Locked, Protected, ReadProtected}
```

CheckTag

`CheckTag`

CheckTag specifies what CellTag to check for prior to dividing a cell with DivideCellAtLineBreaks. If this CellTag is n CellTags then the cell is not divided. The default value of CheckTag is None which specifies that the cell should be d values of its CellTags.

Attributes for `CheckTag`

```
{Locked, Protected, ReadProtected}
```

CleanReissPieces

`CleanReissPieces`

CleanReissPieces[nb] cleans out all left over Reiss's Pieces from the notebook nb.

Attributes for `CleanReissPieces`

```
{Locked, Protected, ReadProtected}
```

CleanseDiariesNotebooksAndPackagesDatabase

`CleanseDiariesNotebooksAndPackagesDatabase`

CleanseDiariesNotebooksAndPackagesDatabase[] removes entries from DiariesNotebooksAndPackagesDatabase if they longer exist at their specified location. If the file in question is on a volume that is note currently mounted then it is ig entries are removed. CleanseDiariesNotebooksAndPackagesDatabase returns a two element list of integers. The first that were removed because the file is no longer located at the specified location. The second is the number of redund removed.

Attributes for `CleanseDiariesNotebooksAndPackagesDatabase`

```
{Locked, Protected, ReadProtected}
```

CleanUpDashboardNotebooks

CleanUpDashboardNotebooks

CleanUpDashboardNotebooks[] removes any items from $DashboardNotebooks that are not NotebookObjects and also NotebookObjects that are not open.

Attributes for `CleanUpDashboardNotebooks`

```
{Locked, Protected, ReadProtected}
```

ClearAttributesString

ClearAttributesString

ClearAttributesString["form", attr] clears attr from the list of attributes of all symbols whose names match any of the stri

Attributes for `ClearAttributesString`

```
{Locked, Protected, ReadProtected}
```

ClearDefaultDiaryDirectory

ClearDefaultDiaryDirectory

ClearDefaultDiaryDirectory[] clears the value of $DefaultDiaryDirectory and resets it to "None".

Attributes for `ClearDefaultDiaryDirectory`

```
{Locked, Protected, ReadProtected}
```

ClearFavoriteDiaries

ClearFavoriteDiaries

ClearFavoriteDiaries[] clears the list of Diary favorites, $FavoriteDiaries and refreshes the FavoritesAndRecentPalette i

Attributes for `ClearFavoriteDiaries`

```
{Locked, Protected, ReadProtected}
```

ClearFavoriteNotebooks

`ClearFavoriteNotebooks`

ClearFavoriteNotebooks[] clears the list of notebook favorites, $FavoriteNotebooks and refreshes the FavoritesAndRece

Attributes for `ClearFavoriteNotebooks`

```
{Locked, Protected, ReadProtected}
```

ClearNotebooksMenu

`ClearNotebooksMenu`

ClearNotebooksMenu[] clears the menu items under the File▹Open Recent menu.

Attributes for `ClearNotebooksMenu`

```
{Locked, Protected, ReadProtected}
```

ClearRecentDiaries

`ClearRecentDiaries`

ClearRecentDiaries[] clears the list of recent diaries, $RecentDiaries and refreshes the FavoritesAndRecentPalette if it is

Attributes for `ClearRecentDiaries`

```
{Locked, Protected, ReadProtected}
```

ClearRecentNotebooks

`ClearRecentNotebooks`

ClearRecentNotebooks[] clears the list of recent notebooks, $RecentNotebooks and refreshes the FavoritesAndRecentPa

Attributes for ClearRecentNotebooks

```
{Locked, Protected, ReadProtected}
```

ClearWorkFlow

ClearWorkFlow

ClearWorkFlow[name] clears the WorkFlow with the given name.

Attributes for ClearWorkFlow

```
{HoldFirst, Locked, Protected, ReadProtected}
```

CloseAllDiaries

CloseAllDiaries

CloseAllDiaries[] saves and closes all open Diary notebooks. The returned value of CloseAllDiaries[] is $CurrentDiaryN

Attributes for CloseAllDiaries

```
{Locked, Protected, ReadProtected}
```

CloseAllPalettes

CloseAllPalettes

CloseAllPalettes[] closes all currently open palettes associated with this package with the exception of the WorkLifeToo
notebooks whose names are of the form Names["Diary`Diary`$*PaletteNotebook"].

Default options for CloseAllPalettes

```
{LeaveOpen → {$WorkLifeToolsPaletteNotebook}, Refresh$OpenPalettes → True}
```

Attributes for CloseAllPalettes

```
{Protected, ReadProtected}
```

CloseDiary

`CloseDiary`

CloseDiary[] saves and closes the current Diary. By default all Cells are locked that are of a style contained in the list $I
CloseDiary[All] locks all cells.

Attributes for `CloseDiary`

```
{Locked, Protected, ReadProtected}
```

CloseOtherDiaries

`CloseOtherDiaries`

CloseOtherDiaries[] saves and closes all open Diary notebooks other than the $CurrentDiaryNotebook. If no $CurrentD
open diaries are closed. The returned value of CloseOtherDiaries[] is $CurrentDiaryNotebook.

Attributes for `CloseOtherDiaries`

```
{Locked, Protected, ReadProtected}
```

Comment

`Comment`

Comment[string,None]

Attributes for `Comment`

```
{HoldAll, Locked, Protected, ReadProtected}
```

ComputationCellOperator

`ComputationCellOperator`

ComputationCellOperator["x"] is the operator used on the data contained in computation cells of type "x". Its default va

Attributes for `ComputationCellOperator`

```
{}
```

ComputationCellOperatorName

`ComputationCellOperatorName`

ComputationCellOperatorName["x"] gives a name associated with the computation operator for cells of type "x".

Attributes for `ComputationCellOperatorName`

```
{}
```

ComputationPalette

`ComputationPalette`

ComputationPalette[] opens the Computation palette.

Attributes for `ComputationPalette`

```
{Locked, Protected, ReadProtected}
```

ComputeDiaryNotebook

`ComputeDiaryNotebook`

ComputeDiaryNotebook[computationCellTagSuffix,style,computationFunction] takes the contents of all cells in the $Cu
have the cell tag "ComputationCell"<>computationCellTagSuffix, applies the function computationFunction to their c
to a cell of cell style "style." For each cell, the contribution to the data that is used is a list containing three elements.
explicitly appears in the contents of the cell. The second element is a list of the CellTags of that Cell. Included in the
is the DiaryDate tag, as well as possibly others. The third element is a list of the TaggingRules of that Cell.

The form ComputeDiaryNotebook[{compCellTagSuffices...},style,computationFunction] takes the data from multiple c
given through the list {compCellTagSuffices...}. The computationFunction should be a function of Length[{compCe

Also note that, if computationCellTagSuffix contains white space, the white space will be automatically removed before

Attributes for `ComputeDiaryNotebook`

```
{Locked, Protected, ReadProtected}
```

ContextQ

`ContextQ`

ContextQ[z] returns True if z is a valid context (a string with no WhiteSpace characters and ending in "`").

Attributes for `ContextQ`

```
{Locked, Protected, ReadProtected}
```

ContractDiaryNotebook

`ContractDiaryNotebook`

ContractDiaryNotebook[] closes all subgroups of the entries in the current Diary except for the most current one.

Attributes for `ContractDiaryNotebook`

```
{Locked, Protected, ReadProtected}
```

ConvertToDiary

`ConvertToDiary`

ConvertToDiary[nb] converts the specified notebook, nb which should be a NotebookObject, into a Diary. The noteboo
executing ConvertToDiary. ConvertToDiary automatically saves the resulting converted notebook, overwriting the o
that the original be backed up prior to executing ConvertToDiary. ConvertToDiary sets the resulting notebook to be t
$CurrentDiaryNotebook, its file to be the $CurrentDiaryNotebookFile and its directory to be the $CurrentDiaryNoteb

Attributes for `ConvertToDiary`

```
{Locked, Protected, ReadProtected}
```

ConvertToDiaryDirectory

`ConvertToDiaryDirectory`

ConvertToDiaryDirectory["directory"] converts the directory into a Diary directory if it is not one already. This involve
"Notebooks", "Databases", "Packages", "OtherFiles", and "Blogs" subdirectories. Then all Packages are moved into tl
and all Notebooks are moved into the "Notebooks" subdirectory. Any remaining non-Mathematica files are moved ir
subdirectory. All Notebook are then processes by opening them and appropriately internally tagging them and registe
DiariesNotebooksAndPackagesDatabase before resaving and closing them. ConvertToDiaryDirectory[] opens a file s
can choose any file in the directory that you want to convert. Once that file is chosen, ConvertToDiaryDirectory will
described above. If none of the notebooks in the directory are Diaries, ConvertToDiaryDirectory opens a dialog that a
diary in the directory so that there is at least one Diary in the directory.

Attributes for `ConvertToDiaryDirectory`

```
{Locked, Protected, ReadProtected}
```

CopyBlogToWebDirectory

CopyBlogToWebDirectory

CopyBlogToWebDirectory[blogName,{webSiteLocalPath, blogWebName}] moves a copy of the blog to a local directo
upload it to a web site. blogName is the name of the Blog. webSiteLocalPath is the full path to the local version of yo
blogWebName is the name of the subdirectory of your web site's local root directory in which the Blog resides. All of
CopyBlogToWebDirectory should be strings. The form CopyBlogToWebDirectory[blogName,{webSiteLocalPath}]
be put into the directory, webSiteLocalPath.

Attributes for `CopyBlogToWebDirectory`

```
{Locked, Protected, ReadProtected}
```

CopyFileToOtherFilesDirectory

CopyFileToOtherFilesDirectory

CopyFileToOtherFilesDirectory[] copies a selected file (chosen via a dialog) to the current Diary's OtherFiles Directory

Default options for `CopyFileToOtherFilesDirectory`

```
{AutoDelete → False}
```

Attributes for `CopyFileToOtherFilesDirectory`

```
{Protected, ReadProtected}
```

CopyNotebookToNotebooksDirectory

`CopyNotebookToNotebooksDirectory`

CopyNotebookToNotebooksDirectory[] copies a selected notebook (chosen via a dialog) to the current Diary's Noteboo

Default options for `CopyNotebookToNotebooksDirectory`

```
{AutoDelete → False}
```

Attributes for `CopyNotebookToNotebooksDirectory`

```
{Protected, ReadProtected}
```

CreateComputationEntry

`CreateComputationEntry`

CreateComputationEntry[computationCellTagSuffix] creates a cell in $CurrentDiaryNotebook to add computation data.
cell tag that is the concatenation of the strings "ComputationCell" and computationCellTagSuffix. The cells in the $C
this cell tag can be computed using the function ComputeDiaryNotebook. CreateComputationEntry[contents,computa
a cell in $CurrentDiaryNotebook with the contents "contents". Note that "contents" must be of the proper form to be
Expression. Also note that, if computationCellTagSuffix contains white space, the white space will be automatically r

Attributes for `CreateComputationEntry`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

CreateDatabase

`CreateDatabase`

CreateDatabase[name, {records...}, fieldNames] creates a database with the name "name" containing the records, {recor
given by fieldNames, and this must be a list of distinct strings with length equal to the number of fields (columns) in t
is placed in a directory called "name" in the Database subdirectory of the current Diary directory. To create a new dat
CreateDatabase[name, data, fieldNames] with data containing a single record. Alternatively you can use CreateDataba
create a database with zero records.

Default options for `CreateDatabase`

```
{IndexedFields → All, Directory → Automatic, FieldTypes → Automatic}
```

Attributes for `CreateDatabase`

```
{HoldFirst, Protected, ReadProtected}
```

CreateDatabaseFromVCards

`CreateDatabaseFromVCards`

CreateDatabaseFromVCards[name,file] creates a database with the specified name from vCards contained in the specifi

Attributes for `CreateDatabaseFromVCards`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

CreateDiaryDirectory

`CreateDiaryDirectory`

CreateDiaryDirectory["directoryName"]

Attributes for `CreateDiaryDirectory`

```
{Locked, Protected, ReadProtected}
```

CreateDiaryDirectorySubdirectory

`CreateDiaryDirectorySubdirectory`

CreateDiaryDirectorySubdirectory[directoryName,subdirectoryName]

Attributes for `CreateDiaryDirectorySubdirectory`

```
{Locked, Protected, ReadProtected}
```

CreateDiaryHTMLDirectory

`CreateDiaryHTMLDirectory`

CreateDiaryHTMLDirectory[] creates a subdirectory of the current Diary directory. The directory's name is that of the c
with the string HTML appended. This directory is used to store an html "blog" version of the Diary when it is genera

DiaryToSimpleBlog.

Attributes for `CreateDiaryHTMLDirectory`

```
{Locked, Protected, ReadProtected}
```

CreateDiaryNBDirectory

`CreateDiaryNBDirectory`

CreateDiaryNBDirectory[] creates a subdirectory of the current Diary directory. The directory's name is that of the curr
the string NB appended. This directory is used to store a hyperlinked set of notebooks generated by DiaryToHyperlin

Attributes for `CreateDiaryNBDirectory`

```
{Locked, Protected, ReadProtected}
```

CreateDiarySelectionHTMLDirectory

`CreateDiarySelectionHTMLDirectory`

CreateDiarySelectionHTMLDirectory[] creates a subdirectory of the current Diary directory. The directory's name is th
notebook with the string HTMLSel appended. This directory is used to store html versions of the selections from a D
DiarySelectionToHTML.

Attributes for `CreateDiarySelectionHTMLDirectory`

```
{Locked, Protected, ReadProtected}
```

CreateDiaryTemplateFromDiary

`CreateDiaryTemplateFromDiary`

CreateDiaryTemplateFromDiary["name"] creates a new Diary template from the current Diary and gives the template th

Attributes for `CreateDiaryTemplateFromDiary`

```
{Locked, Protected, ReadProtected}
```

CreateDiaryXVDirectory

`CreateDiaryXVDirectory`

CreateDiaryXVDirectory[] creates a subdirectory of the current Diary directory. The directory's name is that of the curr
the string XV appended. This directory is used to store an archive version of the Diary when it is generated by Archi

Attributes for `CreateDiaryXVDirectory`

`{Locked, Protected, ReadProtected}`

CreateDirectoryDialog

`CreateDirectoryDialog`

CreateDirectoryDialog[] opens a dialog that allows the creation of a directory that is a subdirectory of $CurrentDiaryNo
CreateDirectoryDialog[dir] allows for the creation of a directory that is a subdirectory of the directory "dir".

Attributes for `CreateDirectoryDialog`

`{Locked, Protected, ReadProtected}`

CreateFormTemplateFromNotebook

`CreateFormTemplateFromNotebook`

CreateFormTemplateFromNotebook["name"] creates a new Form Template from the current InputNotebook based on th
in that notebook and gives the template the name "name." CreateFormTemplateFromNotebook[nb,"name"] creates a t
nb.

Attributes for `CreateFormTemplateFromNotebook`

`{Locked, Protected, ReadProtected}`

CreateHyperlink

`CreateHyperlink`

CreateHyperlink[] opens the "Create Hyperlink" dialog.

Attributes for `CreateHyperlink`

```
{Locked, Protected, ReadProtected}
```

CreateNewDatabaseDirectory

`CreateNewDatabaseDirectory`

CreateNewDatabaseDirectory[name] creates a new database directory that is a subdirectory to the Database subdirectory
directory.

Attributes for `CreateNewDatabaseDirectory`

```
{Locked, Protected, ReadProtected}
```

CreateNotebookFromTaggedCells

`CreateNotebookFromTaggedCells`

CreateNotebookFromTaggedCells[nb,tag] creates a new notebook containing those cells from nb that are tagged with th
new notebook has the same notebook options as nb. However, new options can be supplied to the new notebook as op
CreateNotebookFromTaggedCells.

Default options for `CreateNotebookFromTaggedCells`

```
{SaveBackupToolbarCell → True}
```

Attributes for `CreateNotebookFromTaggedCells`

```
{Protected, ReadProtected}
```

CreateOrganization

`CreateOrganization`

CreateOrganization["organization",{"ruleName1":→rule1,...}] creates an Organization with the name "organization" and
which must be in the form of delayed rules). CreateOrganization["organization"] creates the Organization but specifi

Attributes for `CreateOrganization`

```
{Locked, Protected, ReadProtected}
```

CreateSlideShowFromDiary

CreateSlideShowFromDiary

CreateSlideShowFromDiary[] creates a slide show from the current Diary notebook. Each Section of the current Diary i
slide.

Default options for CreateSlideShowFromDiary

```
{SaveSlideShow → True}
```

Attributes for CreateSlideShowFromDiary

```
{Protected, ReadProtected}
```

CreateSlideShowFromNotebook

CreateSlideShowFromNotebook

CreateSlideShowFromNotebook[nb] creates a slide show from the notebook nb. Each Section of the notebook is assign
CreateSlideShowFromNotebook[nb, cellType] creates a slide show from the notebook nb using cells grouped under c
assigned to a separate slide.

Default options for CreateSlideShowFromNotebook

```
{SaveSlideShow → True}
```

Attributes for CreateSlideShowFromNotebook

```
{Protected, ReadProtected}
```

CreateTimeTaggedEntry

CreateTimeTaggedEntry

CreateTimeTaggedEntry[] creates a new time tagged entry at the end of the current Diary notebook given by $CurrentD
CreateTimeTaggedEntry[contents, style] creates an entry with the contents of the cell style "style." CreateTimeTagge
entry of cell type "style" where the contents of the cell is the current date.

Default options for CreateTimeTaggedEntry

```
{DateOrder :→ $DateOrder}
```

Attributes for `CreateTimeTaggedEntry`

```
{HoldFirst, Protected, ReadProtected}
```

CreateTimeTaggedEntryAt

CreateTimeTaggedEntryAt

CreateTimeTaggedEntryAt[] creates a new time tagged entry after the currently selected cell of the current Diary notebo
$CurrentDiaryNotebook.CreateTimeTaggedEntryAt[contents, style] creates an entry with the contents of the cell styl
"contents" must be of the proper form to be the first argument of a Cell Expression. CreateTimeTaggedEntryAt[style]
type "style" where the contents of the cell is the current date.

Default options for `CreateTimeTaggedEntryAt`

```
{DateOrder :→ $DateOrder}
```

Attributes for `CreateTimeTaggedEntryAt`

```
{Protected, ReadProtected}
```

CreateTimeTaggedEntryCheckSectionHeading

CreateTimeTaggedEntryCheckSectionHeading

CreateTimeTaggedEntryCheckSectionHeading[contents,CellStyle] creates a new time tagged entry at the end of the cur
by $CurrentDiaryNotebook. The entry is preceded by default with a Section cell with the current date information if s
created in the Diary for the current day. Note that "contents" must be of the proper form to be the first argument of a C
CreateTimeTaggedEntryCheckSectionHeading["style"] does the same for a cell of cell style "style."

Default options for `CreateTimeTaggedEntryCheckSectionHeading`

```
{DateOrder :→ $DateOrder}
```

Attributes for `CreateTimeTaggedEntryCheckSectionHeading`

```
{HoldFirst, Protected, ReadProtected}
```

CreateToDo

CreateToDo

CreateToDo["text",n] creates an entry in the current Diary notebook that is marked as a "ToDo" item with priority n Wit chosen from Range[7]}.

Attributes for `CreateToDo`

```
{Locked, Protected, ReadProtected}
```

CreateWorkFlow

`CreateWorkFlow`

CreateWorkFlow[name, {"flowname1":→flow1, "flowname2":→flow2,... }] creates a WorkFlow with the given name bas
rules must be in the form of Delayed Rules. If a previous WorkFlow was created with the same name, it will be replac

Attributes for `CreateWorkFlow`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

CurrentDiary

`CurrentDiary`

CurrentDiary[Notebook] gives the current Diary notebook object. CurrentDiary[FileName] gives the current Diary notel
name.CurrentDiary[Directory] gives the current Diary notebook directory. CurrentDiary[] gives a list of these three it

Attributes for `CurrentDiary`

```
{Locked, Protected, ReadProtected}
```

CurrentDiaryBackups

`CurrentDiaryBackups`

CurrentDiaryBackups[] gives a list of backups that have been made for the current Diary. It is presented as a list where e
first member of the pair is the date in the current computer's time zone when the backup was made. The second meml
name of the corresponding backup. CurrentDiaryBackups[TabularReport] produces a formatted table with buttons tha
backup.

Attributes for `CurrentDiaryBackups`

```
{Locked, Protected, ReadProtected}
```

CurrentDiaryNotebookDirectory

CurrentDiaryNotebookDirectory

CurrentDiaryNotebookDirectory[] gives the value of $CurrentDiaryNotebookDirectory.

Attributes for `CurrentDiaryNotebookDirectory`

```
{Locked, Protected, ReadProtected}
```

CurrentDiaryNotebookOpenQ

CurrentDiaryNotebookOpenQ

CurrentDiaryNotebookOpenQ[] gives True if the Current Diary notebook is open and False if it is closed. If no Diary is
$CurrentDiaryNotebook=None) then CurrentDiaryNotebookOpenQ[] returns unevaluated.

Attributes for `CurrentDiaryNotebookOpenQ`

```
{Locked, Protected, ReadProtected}
```

CurrentDirectoriesAndFiles

CurrentDirectoriesAndFiles

CurrentDirectoriesAndFiles[] opens a notebook with information on some of the current directories and files associated

Attributes for `CurrentDirectoriesAndFiles`

```
{Locked, Protected, ReadProtected}
```

CurrentlyOpenNotebooksByIndex

CurrentlyOpenNotebooksByIndex

CurrentlyOpenNotebooksByIndex[TabularReport] gives a list of all currently open notebooks and their index in that list
of the form {NotebookObject[fe, id], n} where n is the position in the list. CurrentlyOpenNotebooksByIndex[n] gives
the list. CurrentlyOpenNotebooksByIndex[TableForm] is equivalent to TableForm[CurrentlyOpenNotebooksByIndex
CurrentlyOpenNotebooksByIndex[Notebook] produces the list in a new notebook. CurrentlyOpenNotebooksByIndex
without a formatted table.

Attributes for `CurrentlyOpenNotebooksByIndex`

```
{Locked, Protected, ReadProtected}
```

Custom1Palette

`Custom1Palette`

Custom1Palette[] opens the first custom Palette.

Attributes for `Custom1Palette`

```
{Locked, Protected, ReadProtected}
```

Custom2Palette

`Custom2Palette`

Custom2Palette[] opens the second custom Palette.

Attributes for `Custom2Palette`

```
{Locked, Protected, ReadProtected}
```

Custom3Palette

`Custom3Palette`

Custom3Palette[] opens the third custom Palette.

Attributes for `Custom3Palette`

```
{Locked, Protected, ReadProtected}
```

Custom4Palette

`Custom4Palette`

Custom4Palette[] opens the fourth custom Palette.

Attributes for `Custom4Palette`

```
{Locked, Protected, ReadProtected}
```

Custom5Palette

`Custom5Palette`

Custom5Palette[] opens the fifth custom Palette.

Attributes for `Custom5Palette`

```
{Locked, Protected, ReadProtected}
```

Custom6Palette

`Custom6Palette`

Custom6Palette[] opens the sixth custom Palette.

Attributes for `Custom6Palette`

```
{Locked, Protected, ReadProtected}
```

Dashboard

`Dashboard`

Dashboard[] opens the dashboard.

Default options for `Dashboard`

```
{RefreshDashboard → True}
```

Attributes for `Dashboard`

```
{Protected, ReadProtected}
```

DashboardElements

DashboardElements

DashboardElements[] gives a list of the Dashboard elements.

Attributes for `DashboardElements`

```
{Locked, Protected, ReadProtected}
```

DatabaseDirectory

DatabaseDirectory

DatabaseDirectory[] gives the database directory in the current Diary directory. DatabaseDirectory[name] gives the dire
"name" resides.

Attributes for `DatabaseDirectory`

```
{Locked, Protected, ReadProtected}
```

DatabaseFieldNames

DatabaseFieldNames

DatabaseFieldNames[name] gives the list of names of the fields in the given database if it has been loaded. DatabaseFie
ith field name of the database. If i>RecordLength[name] then DatabaseFieldNames[name,i] returns $Failed.

Attributes for `DatabaseFieldNames`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

DatabaseFieldTypes

DatabaseFieldTypes

DatabaseFieldTypes[name] gives the list of patterns that items within records for the given database must match. The le
as the number of DatabaseFieldNames. DatabaseFieldTypes[name,i] gives the ith pattern that the ith item in a record
i>RecordLength[name] then DatabaseFieldTypes[name,i] returns _?False. Field types for a database are specified thr
FieldTypes to CreateDatabase. You can change the values of DatabaseFieldTypes through the function ChangeDatab

Attributes for `DatabaseFieldTypes`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

DatabaseFile

`DatabaseFile`

DatabaseFile["directory"] gives the full path to a database file if it exists within the directory. If there is no such file in tl
None is returned. A database file is one of the form "*DB.m".

Attributes for `DatabaseFile`

```
{Locked, Protected, ReadProtected}
```

DatabaseFileInDirectoryQ

`DatabaseFileInDirectoryQ`

DatabaseFileInDirectoryQ["directory"] determines whether a database file exists within the directory. A database file is

Attributes for `DatabaseFileInDirectoryQ`

```
{Locked, Protected, ReadProtected}
```

DatabaseFind

`DatabaseFind`

DatabaseFind[name,item,"fieldName"] looks for item in the field of the database given by name with the field name "fie
database records that have a match in that field. DatabaseFind[name,item,j] looks for item in the jth field of the databa
DatabaseFind[name,pattern] looks for matches to the pattern in the form of a list that is NumberOfDatabaseFields[nai
DatabaseFind[name, Function|DatabasePattern] looks for database records according to a pure function or a Database

Default options for `DatabaseFind`

```
{MatchingCondition → None, GenerateNotebook → False}
```

Attributes for `DatabaseFind`

```
{HoldFirst, Protected, ReadProtected}
```

DatabasePattern

DatabasePattern

DatabasePattern[recordPattern] represents a pattern for searching database records using DatabaseFieldNames.

Attributes for `DatabasePattern`

```
{HoldAll, Locked, Protected, ReadProtected}
```

Databases

Databases

Databases[] gives a list of the databases in the current Diary's Database directory. The list is in the same form as the list
It is in the form of a list with each element a list of length two. The first element of each entry is the name of the datab
the full path name to the database. The databases that the package currently knows about, excluding the default databa
executing Databases[All]. The default databases in the package database directory can be listed by executing Databas
databases are not listed.

Attributes for `Databases`

```
{Locked, Protected, ReadProtected}
```

DatabasesPalette

DatabasesPalette

DatabasesPalette[] opens the Database Palette.

Attributes for `DatabasesPalette`

```
{Locked, Protected, ReadProtected}
```

DateFromDiaryDateTag

DateFromDiaryDateTag

DateFromDiaryDateTag[tag] extracts the date from the given DiaryDate tag. This tag must either be a string that matche
expression of this form.

Attributes for `DateFromDiaryDateTag`

```
{Locked, Protected, ReadProtected}
```

DateOrder

`DateOrder`

DateOrder is an option to functions that display a date that determine how the ordering of Year, Month, and Day appear possibilities are Automatic, "North American", and "European". "North American" gives a date in the month/day/yea gives the date in the day/month/year format. Automatic is the same as "North American". "MonthDay" is equivalent "DayMonth" is equivalent to "European".

Attributes for `DateOrder`

```
{Locked, Protected, ReadProtected}
```

DateQ

`DateQ`

DateQ[x] is true if x can represent a date in the form given by Date[].

Attributes for `DateQ`

```
{Locked, Protected, ReadProtected}
```

DateStringFromTag

`DateStringFromTag`

DateStringFromTag[tag] gives a conventional calendar date from a tag string.

Default options for `DateStringFromTag`

```
{DateOrder :→ $DateOrder, TimeForm → 12, IncludeDayOfTheWeek → True,
 IncludeTime → False, IncludeSeconds → True, OriginatingTimeZone → 0}
```

Attributes for `DateStringFromTag`

```
{Protected, ReadProtected}
```

DateTagCell

`DateTagCell`

DateTagCell[notebook] adds a DiaryDate tag to the currently selected cell in notebook if it doesn't have one already. Da
date tags the currently selected cell replacing any prior DiaryDate tag if necessary. DateTagCell[] is equivalent to
DateTagCell[$CurrentDiaryNotebook] and DateTagCell[All] is equivalent to DateTagCell[$CurrentDiaryNotebook,A
chosen then DateTagCell only tags the first one.

Attributes for `DateTagCell`

```
{Locked, Protected, ReadProtected}
```

DecryptNotebook

`DecryptNotebook`

DecryptNotebook["file","password"] decrypts a notebook that was encrypted with EncryptNotebook using the given pas
an encrypted notebook to someone who does not have A WorkLife FrameWork™ then ExportableDecryptingFunctio
that you can give them that opens a notebook with a function DecryptEncryptedNotebook that can be used by individ
access to the WorkLife FrameWork™ Package to decrypt notebooks that have been encrypted using EncryptNoteboo

Attributes for `DecryptNotebook`

```
{Locked, Protected, ReadProtected}
```

DecryptNotebookDialog

`DecryptNotebookDialog`

DecryptNotebookDialog[] opens the DecryptNotebookDialog window.

Attributes for `DecryptNotebookDialog`

```
{Locked, Protected, ReadProtected}
```

DefaultCodeCell

`DefaultCodeCell`

DefaultCodeCell[] is a default code cell.

Attributes for `DefaultCodeCell`

```
{Locked, Protected, ReadProtected}
```

DefaultDatabaseFieldValues

`DefaultDatabaseFieldValues`

DefaultDatabaseFieldValues is an option to AddDatabaseFields that determines how values to new fields are to be assig
records in the database. Its default value is GUID (or Automatic) which assigns a unique integer to the new fields in t
other possibility is for DefaultDatabaseFieldValues to be a list of pure functions of one variable. That variable is the
original record.

Attributes for `DefaultDatabaseFieldValues`

```
{Locked, Protected, ReadProtected}
```

DefaultDiaryDirectory

`DefaultDiaryDirectory`

DefaultDiaryDirectory[] gives the value of $DefaultDiaryDirectory.

Attributes for `DefaultDiaryDirectory`

```
{Locked, Protected, ReadProtected}
```

DefaultParameterValue

`DefaultParameterValue`

DefaultParameterValue[param] gives the default value of the parameter if it has a value and if it is in the Diary.m packa

Attributes for `DefaultParameterValue`

```
{HoldAll, Locked, Protected, ReadProtected}
```

DeleteBackupsFromDirectoryList

DeleteBackupsFromDirectoryList

DeleteBackupsFromDirectoryList[{directories...}] returns from the list of directories only those that are not backup dire
Also, if any files are included in the list they too are removed.

Attributes for `DeleteBackupsFromDirectoryList`

```
{Locked, Protected, ReadProtected}
```

DeleteBackupsFromNotebookList

DeleteBackupsFromNotebookList

DeleteBackupsFromNotebookList[{files...}] returns from the list of files only those that are (1) Mathematica notebooks
files of other files.

Attributes for `DeleteBackupsFromNotebookList`

```
{Locked, Protected, ReadProtected}
```

DeleteCellStylesFromDiaryEntryPalette

DeleteCellStylesFromDiaryEntryPalette

DeleteCellStylesFromDiaryEntryPalette["CellStyle"] removes the CellStyle to the Diary Entry palette.

Attributes for `DeleteCellStylesFromDiaryEntryPalette`

```
{Locked, Protected, ReadProtected}
```

DeleteCellStylesFromEssayPalette

DeleteCellStylesFromEssayPalette

DeleteCellStylesFromEssayPalette["CellStyle"] removes the CellStyle to the Essay palette.

Attributes for DeleteCellStylesFromEssayPalette

```
{Locked, Protected, ReadProtected}
```

DeleteCellTag

DeleteCellTag

DeleteCellTag[nb,"tag"] removes the tag "tag" from the CellTags of the currently selected cell in the notebook nb if it is CellTags. DeleteCellTag[nb,"tag",Head] removes all instances of a tag of the form "tag[*]" from the currently selecte DeleteCellTag[nb,All] removes all cell tags from the currently selected cell in the notebook nb. DeleteCellTag[nb,All from a single Cell at a time. The other forms can act on a selection of multiple Cells.

Attributes for DeleteCellTag

```
{Locked, Protected, ReadProtected}
```

DeleteCellTaggingRule

DeleteCellTaggingRule

DeleteCellTaggingRule[nb,tag] deletes "tag" from the TaggingRules for the selected cell in the notebook object nb. tag

Attributes for DeleteCellTaggingRule

```
{HoldRest, Locked, Protected, ReadProtected}
```

DeleteCurrentPackageFromPluginsDirectory

DeleteCurrentPackageFromPluginsDirectory

DeleteCurrentPackageFromPluginsDirectory[] deletes any copy of the current package from the Plugins directory. The resides in its original place in PackagesDirectory[].

Attributes for DeleteCurrentPackageFromPluginsDirectory

```
{Locked, Protected, ReadProtected}
```

DeleteDashboardElements

DeleteDashboardElements

DeleteDashboardElements[{"elements"...}] deletes the elements from the Dashboard.

Attributes for `DeleteDashboardElements`

```
{Locked, Protected, ReadProtected}
```

DeleteDatabase

DeleteDatabase

DeleteDatabase[name] deletes the database name. This should be used with care.

Default options for `DeleteDatabase`

```
{BackupFirst → True}
```

Attributes for `DeleteDatabase`

```
{HoldFirst, Protected, ReadProtected}
```

DeleteDatabaseRecords

DeleteDatabaseRecords

DeleteDatabaseRecords[name,{records...}] deletes the records from the database "name". These records are stored in De
the database is reloaded, whereupon the records are deleted from the database file. Prior to that the deleted records are
in DatabaseDirectory[name].

Attributes for `DeleteDatabaseRecords`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

DeleteDiaryKeywords

DeleteDiaryKeywords

DeleteDiaryKeywords[{"keyword1","keyword2",...}] deletes the keywords from the list of those that the package uses t
directory $CurrentDiaryNotebookDirectory. DeleteDiaryKeywords[All] deletes all possible keywords except for the

Attributes for `DeleteDiaryKeywords`

```
{Locked, Protected, ReadProtected}
```

DeleteDiaryTemplate

`DeleteDiaryTemplate`

DeleteDiaryTemplate["name"] deletes the Diary Template corresponding to "name".

Attributes for `DeleteDiaryTemplate`

```
{Locked, Protected, ReadProtected}
```

DeletedRecords

`DeletedRecords`

DeletedRecords[name] gives a list of the recently deleted records from the database given by name.

Attributes for `DeletedRecords`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

DeleteEmail

`DeleteEmail`

DeleteEmail[nb] deletes the email from the notebook nb. The insertion point must me within the email to be deleted. Th
copied to the clipboard. Set the value of $DeleteEmailCreateNotebook to determine whether or not to create a new no
email.

Attributes for `DeleteEmail`

```
{Locked, Protected, ReadProtected}
```

DeleteEncryptedOriginal

DeleteEncryptedOriginal

DeleteEncryptedOriginal is an option for EncryptNotebook and EncryptNotebookDialog that determines whether the no
encrypted should be deleted once it has been encrypted. Its default value is False.

Attributes for `DeleteEncryptedOriginal`

```
{Locked, Protected, ReadProtected}
```

DeleteEssay

DeleteEssay

DeleteEssay[nb] deletes the Essay from the notebook nb. The insertion point must me within the Essay to be deleted. T
copied to the clipboard. Set the value of $DeleteEssayCreateNotebook to determine whether or not to create a new no
Essay. Note however that, even if $DeleteEssayCreateNotebook is False, the deleted essay will be placed in the Clipb

Attributes for `DeleteEssay`

```
{Locked, Protected, ReadProtected}
```

DeleteFileFrom$FileSetsDialog

DeleteFileFrom$FileSetsDialog

DeleteFileFrom$FileSetsDialog[] opens a dialog that allows you to delete files from named FileSets in $FileSets.

Attributes for `DeleteFileFrom$FileSetsDialog`

```
{Locked, Protected, ReadProtected}
```

DeleteFormattingBackgroundColors

DeleteFormattingBackgroundColors

DeleteFormattingTextColors[{"name",...}] removes the colors with the name "name" from the formatting palette. "name

Attributes for `DeleteFormattingBackgroundColors`

```
{Locked, Protected, ReadProtected}
```

DeleteFormattingTextColors

`DeleteFormattingTextColors`

DeleteFormattingTextColors[{"name",...}] removes the colors with the name "name" from the formatting palette. Each

Attributes for `DeleteFormattingTextColors`

```
{Locked, Protected, ReadProtected}
```

DeleteFormTemplate

`DeleteFormTemplate`

DeleteFormTemplate["name"] deletes the Form Template corresponding to "name".

Attributes for `DeleteFormTemplate`

```
{Locked, Protected, ReadProtected}
```

DeleteFrom$FavoriteDiaries

`DeleteFrom$FavoriteDiaries`

DeleteFrom$FavoriteDiaries[file] deletes the file "file" from the list of favorite diaries.

Attributes for `DeleteFrom$FavoriteDiaries`

```
{Locked, Protected, ReadProtected}
```

DeleteFrom$FavoriteNotebooks

`DeleteFrom$FavoriteNotebooks`

DeleteFrom$FavoriteNotebooks[file] deletes the file "file" from the list of favorite notebooks.

Attributes for `DeleteFrom$FavoriteNotebooks`

```
{Locked, Protected, ReadProtected}
```

DeleteFrom$FileSets

DeleteFrom$FileSets

DeleteFrom$FileSets[name,file] deletes the file from the FileSet with the given name in $FileSets.

Attributes for `DeleteFrom$FileSets`

```
{Locked, Protected, ReadProtected}
```

DeleteFrom$Path

DeleteFrom$Path

DeleteFrom$Path[dir] deletes dir from $Path if it is in it.

Attributes for `DeleteFrom$Path`

```
{Locked, Protected, ReadProtected}
```

DeleteFrom$RecentDiaries

DeleteFrom$RecentDiaries

DeleteFrom$RecentDiaries[file] deletes the file "file" from the list of recent diaries.

Attributes for `DeleteFrom$RecentDiaries`

```
{Locked, Protected, ReadProtected}
```

DeleteFrom$RecentNotebooks

DeleteFrom$RecentNotebooks

DeleteFrom$RecentNotebooks[file] deletes the file "file" from the list of recent notebooks.

Attributes for DeleteFrom$RecentNotebooks

```
{Locked, Protected, ReadProtected}
```

DeleteFrom$RSSFeeds

DeleteFrom$RSSFeeds

DeleteFrom$RSSFeeds["name"]

Attributes for DeleteFrom$RSSFeeds

```
{Locked, Protected, ReadProtected}
```

DeleteFullFileSet

DeleteFullFileSet

DeleteFullFileSet[name] deletes the full FileSet with the given name from $FileSets. A confirmation dialog is put up be

Attributes for DeleteFullFileSet

```
{Locked, Protected, ReadProtected}
```

DeleteGUIDFromNotebook

DeleteGUIDFromNotebook

DeleteGUIDFromNotebook[nb] deletes the GUID tag from the TaggingRules of the notebook object nb if it has one.

Attributes for DeleteGUIDFromNotebook

```
{Locked, Protected, ReadProtected}
```

DeleteHeading

DeleteHeading

DeleteHeading[heading,style] deletes the heading "heading" from the list of headings of style "style".

Attributes for `DeleteHeading`

```
{Locked, Protected, ReadProtected}
```

DeleteNotebookTaggingRule

`DeleteNotebookTaggingRule`

DeleteNotebookTaggingRule[nb,tag] deletes "tag" from the TaggingRules for notebook object nb. tag can be a pattern.

Attributes for `DeleteNotebookTaggingRule`

```
{HoldRest, Locked, Protected, ReadProtected}
```

DeleteRSSFeed

`DeleteRSSFeed`

DeleteFrom$RSSFeeds["name"]

Attributes for `DeleteRSSFeed`

```
{Locked, Protected, ReadProtected}
```

DeleteRSSFeedDialog

`DeleteRSSFeedDialog`

DeleteRSSFeedDialog[] opens the Remove an RSSFeed Dialog so that you can delete an RSS Feed from the RSSFeeds

Attributes for `DeleteRSSFeedDialog`

```
{Locked, Protected, ReadProtected}
```

DeleteSaveBackupToolbarCell

`DeleteSaveBackupToolbarCell`

DeleteSaveBackupToolbarCell[nb] deletes the SaveBackupToolbarCells from the notebook nb. DeleteSaveBackupTool
function for the notebook that it is evaluated in.

Attributes for `DeleteSaveBackupToolbarCell`

```
{Locked, Protected, ReadProtected}
```

DeleteToHideTag

`DeleteToHideTag`

DeleteToHideTag[nb] removes the tag "ToHide" from the CellTags of the currently selected cell in the notebook nb if it
CellTags.

Attributes for `DeleteToHideTag`

```
{Locked, Protected, ReadProtected}
```

DeleteWorkLifeSkin

`DeleteWorkLifeSkin`

DeleteWorkLifeSkin[skinName] deletes the WorkLife skin with the given name. That name, skinName, should be a str
names returned by WorkLifeSkins[].

Attributes for `DeleteWorkLifeSkin`

```
{Locked, Protected, ReadProtected}
```

DiaryAccessPalette

`DiaryAccessPalette`

DiaryAccessPalette[] opens the Diary Access palette.

Attributes for `DiaryAccessPalette`

```
{Locked, Protected, ReadProtected}
```

DiaryBlogsDirectory

DiaryBlogsDirectory

DiaryBlogsDirectory[] gives the directory where the blogs associated with the current Diary are located.

Attributes for DiaryBlogsDirectory

```
{Locked, Protected, ReadProtected}
```

DiaryDate

DiaryDate

DiaryDate is a function head indicating a date tag.

Attributes for DiaryDate

```
{Locked, Protected, ReadProtected}
```

DiaryDateQ

DiaryDateQ

DiaryDateQ[x] gives True if x is a DiaryDate and False otherwise. x is a DiaryDate if it either a string that matches "Di
expression of this form. The argument of the DiaryDate must be a numerical date in the form returned by Date[].

Attributes for DiaryDateQ

```
{Locked, Protected, ReadProtected}
```

DiaryDirectoryFunction

DiaryDirectoryFunction

DiaryDirectoryFunction is an option for NewDiaryNotebook and should be a pure function of one variable. If the NewI
NewDiaryNotebook is NewDirectory→True then DiaryDirectoryFunction shows how to create the Diary Directory's
name. Its default value is #& which gives the directory the same name as the Diary. To give the directory the name "
DiaryDirectoryFunction→("aname"&).

Attributes for DiaryDirectoryFunction

```
{Locked, Protected, ReadProtected}
```

DiaryEntriesPalette

DiaryEntriesPalette

DiaryEntriesPalette[] opens a palette containing links to the entries in the current Diary notebook. (Entries are defined as Section heading.) Clicking on the date button brings you to that entry. The associated "Nb" button pops up a notebook DiaryEntriesPalette[CellStyle] does the same for the cell style (a string) CellStyle. (So DiaryEntriesPalette[] is the sa DiaryEntriesPalette["Section"].)

Attributes for DiaryEntriesPalette

```
{Locked, Protected, ReadProtected}
```

DiaryEntryPalette

DiaryEntryPalette

DiaryEntryPalette[] opens the Diary Entry Palette.

Attributes for DiaryEntryPalette

```
{Locked, Protected, ReadProtected}
```

DiaryFiles

DiaryFiles

DiaryFiles[] gives a list of the Diary files in the current default Diary directory with the full directory path information i such files should by default be named with the at least one of the Diary Keywords in each of their names: the default l other Diary Keywords have been added with AddDiaryKeywords then those keywords may also be used in the file na gives a list without the directory information included in the strings.

Attributes for DiaryFiles

```
{Locked, Protected, ReadProtected}
```

DiaryFunctions

DiaryFunctions

DiaryFunctions[] produces a notebook with a table of the functions from the Diary package. DiaryFunctions[stringPatter
with functions that match stringPattern. Thus, DiaryFunctions["*"] is equivalent to DiaryFunctions[].

Attributes for DiaryFunctions

```
{Locked, Protected, ReadProtected}
```

DiaryHeadings

DiaryHeadings

DiaryHeadings[style] gives a list of headings for Diary cells that are of the type "style". "style" can be "Section", "Subs
"Subsubsection". DiaryHeadings[All] give a list of lists of the "Section", "Subsection", and "Subsubsection" headings

Attributes for DiaryHeadings

```
{Locked, Protected, ReadProtected}
```

DiaryHeadingsPalette

DiaryHeadingsPalette

DiaryHeadingsPalette[] opens the Diary headings Palette.

Attributes for DiaryHeadingsPalette

```
{Locked, Protected, ReadProtected}
```

DiaryHTMLDirectory

DiaryHTMLDirectory

DiaryHTMLDirectory[] gives the directory where files resulting from DiaryToSimpleBlog are stored. Note that, althou
gives a result based on the current Diary notebook, the directory may not exist if it hasn't been created by DiaryToSim
functions.

Attributes for `DiaryHTMLDirectory`

```
{Locked, Protected, ReadProtected}
```

DiaryKeywords

`DiaryKeywords`

DiaryKeywords[] gives the list of current Diary keywords.

Attributes for `DiaryKeywords`

```
{Locked, Protected, ReadProtected}
```

DiaryListPalette

`DiaryListPalette`

DiaryListPalette[] opens a palette of the current diaries in the $CurrentDiaryNotebookDirectory. All such files should by
the word "Diary" in each of their names.

Attributes for `DiaryListPalette`

```
{Locked, Protected, ReadProtected}
```

DiaryNBDirectory

`DiaryNBDirectory`

DiaryNBDirectory[] gives the directory where files resulting from DiaryToHyperlinkedNotebook are stored. Note that,
DiaryNBDirectory gives a result based on the current Diary notebook, the directory may not exist if it hasn't been crea
DiaryToHyperlinkedNotebook or similar functions.

Attributes for `DiaryNBDirectory`

```
{Locked, Protected, ReadProtected}
```

DiaryNotebookParametersAndFunctions

`DiaryNotebookParametersAndFunctions`

DiaryNotebookParametersAndFunctions[] gives the list of the names of the Diary notebook parameters and functions w
their defaults by executing ResetDiaryNotebookDefaults[].

Attributes for `DiaryNotebookParametersAndFunctions`

```
{Locked, Protected, ReadProtected}
```

DiaryNotebooksDirectory

`DiaryNotebooksDirectory`

DiaryNotebooksDirectory[] gives the directory where notebooks associated with the current Diary are stored.

Attributes for `DiaryNotebooksDirectory`

```
{Locked, Protected, ReadProtected}
```

DiaryOtherFilesDirectory

`DiaryOtherFilesDirectory`

DiaryOtherFilesDirectory[] gives the directory where other files associated with the current Diary are stored.

Attributes for `DiaryOtherFilesDirectory`

```
{Locked, Protected, ReadProtected}
```

DiaryPluginsDirectory

`DiaryPluginsDirectory`

DiaryPluginsDirectory[] gives the value for $DiaryPluginsDirectory.

Attributes for `DiaryPluginsDirectory`

```
{Locked, Protected, ReadProtected}
```

DiaryQ

`DiaryQ`

DiaryQ[nb] is True if the notebook nb is a Diary. DiaryQ["file"] is True if the file "file" is a Diary file.

Attributes for `DiaryQ`

```
{Locked, Protected, ReadProtected}
```

DiaryScratchNotebooksDirectory

`DiaryScratchNotebooksDirectory`

DiaryScratchNotebooksDirectory[] gives the directory where notebooks associated with the current Diary are stored.

Attributes for `DiaryScratchNotebooksDirectory`

```
{Locked, Protected, ReadProtected}
```

DiaryTemplates

`DiaryTemplates`

DiaryTemplates[] gives a list of the available Diary templates. To create a new template use CreateDiaryTemplateFrom

Attributes for `DiaryTemplates`

```
{Locked, Protected, ReadProtected}
```

DiaryTemplatesPalette

`DiaryTemplatesPalette`

DiaryTemplatesPalette[] opens the Diary templates palette.

Attributes for `DiaryTemplatesPalette`

```
{Locked, Protected, ReadProtected}
```

DiaryToHyperlinkedNotebook

DiaryToHyperlinkedNotebook

DiaryToHyperlinkedNotebook[] creates a Notebook with hyperlinks to individual notebooks each containing one time s
notebook and the associated notebooks that it is hyperlinked to will reside in a directory that is a subdirectory to
$CurrentDiaryNotebookDirectory. The directory's name will be the same as $CurrentDiaryNotebookFile with the ".nl
the characters "NB" appended to the end. DiaryToHyperlinkedNotebook[n] does this for the most recent entries beyo
DiaryToHyperlinkedNotebook[All] is the same as DiaryToHyperlinkedNotebook[]. DiaryToHyperlinkedNotebook[U
directory to include the entries that have been made since the last time DiaryToHyperlinkedNotebook[Update] was ex
the Diary notebook were edited since the last update, these won't be included in the update. To include these, execute
DiaryToHyperlinkedNotebook[All].

Attributes for `DiaryToHyperlinkedNotebook`

```
{Locked, Protected, ReadProtected}
```

DiaryToSimpleBlog

DiaryToSimpleBlog

DiaryToSimpleBlog[] creates a Web page with hyperlinks to web pages embodying the html versions individual notebo
time stamped Diary entry. This web page and the associated web pages that it is hyperlinked to will reside in a directo
$CurrentDiaryNotebookDirectory. The directory's name will be the same as $CurrentDiaryNotebookFile with the ".nl
the characters "HTML" appended to the end. DiaryToSimpleBlog[n] does this for the most recent entries beyond the
DiaryToSimpleBlog[All] is the same as DiaryToSimpleBlog[]. DiaryToSimpleBlog[Update] updates the directory to
have been made since the last time DiaryToSimpleBlog[Update] was executed. If earlier entries to the Diary noteboo
update, these won't be included in the update. To include these, execute DiaryToSimpleBlog[All].

Default options for `DiaryToSimpleBlog`

```
{BlogState → Automatic, Reverse → True}
```

Attributes for `DiaryToSimpleBlog`

```
{Protected, ReadProtected}
```

DiaryXVDirectory

DiaryXVDirectory

DiaryXVDirectory[] gives the directory where files resulting from Archive are stored. Note that, although DiaryXVDir
on the current Diary notebook, the directory may not exist if it hasn't been created by Archive or similar functions.

Attributes for `DiaryXVDirectory`

```
{Locked, Protected, ReadProtected}
```

Directories

Directories

Directories[] gives a list of the directories contained in the current working directory. By default backup directories are r
backup directories set the option ShowBackups→False.

Default options for `Directories`

```
{ShowBackups → False}
```

Attributes for `Directories`

```
{Protected, ReadProtected}
```

DirectoriesInDirectory

DirectoriesInDirectory

DirectoriesInDirectory[dir] gives a list of directories that are in the directory dir.

Attributes for `DirectoriesInDirectory`

```
{Locked, Protected, ReadProtected}
```

DirectoryBrowser

DirectoryBrowser

DirectoryBrowser[] opens an interactive dialog that allows you to select a default Diary directory. DirectoryBrowser[{cl
directories that begin with any of the characters in the list chars.

Attributes for `DirectoryBrowser`

```
{Locked, Protected, ReadProtected}
```

Done

Done

Done is a function head indicating a date tag for a Done. It is also used as a cell tag for items that have been marked as a of a Done is the date when the item was completed. The second argument of a Done is the date when the item was or third argument of a Done is the priority of the original item that was marked Done. If a Done has only a single argum the date when the item was marked Done.

Attributes for `Done`

```
{Locked, Protected, ReadProtected}
```

DoneStatistics

DoneStatistics

DoneStatistics[] generates a list of the number of Dones of each former priority in the current Diary notebook. DoneStat notebook with the data along with a bar graph of the same.

Attributes for `DoneStatistics`

```
{Locked, Protected, ReadProtected}
```

DropReissPiece

DropReissPiece

DropReissPiece[nb] drops a Reiss Piece in the notebook nb. The returned value is a GUID.

Default options for `DropReissPiece`

```
{OtherTags → {}, CellStyle → Text}
```

Attributes for `DropReissPiece`

```
{Protected, ReadProtected}
```

DueDate

DueDate

DueDate is the head of a function indicating the date that a ToDo is due. It appears in the CellTags of a ToDo if it was e
ToDosEntryDialog or the MarkToDoEntryDialog.

Attributes for `DueDate`

```
{Locked, Protected, ReadProtected}
```

EmailAddresses

EmailAddresses

EmailAddresses[file,headerType] gives the email addresses from the mailbox file that are contained in the given type of
headers should be strings and are generally one of: "to", "from", "cc", "bcc". EmailAddresses[file,{headerTypes...}] g
contained in a multiplicity of header types. EmailAddresses[{files...},headerType] gives the email addresses containe
and the given header type. EmailAddresses[{files...},{headerTypes...}] gives the email addresses contained in a mult
types.

Default options for `EmailAddresses`

```
{EmailAliasRules → {}, MailBoxType → Other}
```

Attributes for `EmailAddresses`

```
{Protected, ReadProtected}
```

EmailAliasRules

EmailAliasRules

EmailAliasRules is a option for EmailNetwork and EmailAddresses that specifies a list of rules between email addresses
the same as one another.

Attributes for `EmailAliasRules`

```
{Locked, Protected, ReadProtected}
```

EmailFind

`EmailFind`

EmailFind[notebookGUID,emailGUID] finds the email with the given emailGUID in the notebook with the given noteb
A WorkLife FrameWork.

Attributes for `EmailFind`

```
{Locked, Protected, ReadProtected}
```

EmailFindDialog

`EmailFindDialog`

EmailFindDialog[] opens the email find dialog.

Attributes for `EmailFindDialog`

```
{Locked, Protected, ReadProtected}
```

EmailHeaders

`EmailHeaders`

EmailHeaders["file"] returns the email headers for the emails in the specified mailbox file. The file must be in mbox for
used by Eudora, Thunderbird, and several other email programs.

Default options for `EmailHeaders`

```
{MailBoxType → Other}
```

Attributes for `EmailHeaders`

```
{Protected, ReadProtected}
```

EmailNetwork

`EmailNetwork`

EmailNetwork[{files...}] generates a network graph of the network of emails in the supplied mailbox files. EmailNetwo
single file. EmailNetwork[{files...},n] generates the network graph for the n most populous emails in the files. Email
for a single file.

Default options for `EmailNetwork`

```
{EmailNetworkPlotType → GraphPlot, ShowEmailLabels → False, EmailAliasRules → {}}
```

Attributes for `EmailNetwork`

```
{Protected, ReadProtected}
```

EmailNetworkPlotType

`EmailNetworkPlotType`

EmailNetworkPlotType is an option for EmailNetwork that specifies the sort of network graph to produce. The two poss
TreePlot.

Attributes for `EmailNetworkPlotType`

```
{Locked, Protected, ReadProtected}
```

EmailPalette

`EmailPalette`

EmailPalette[] opens up the Email palette. Additional buttons can be appended to the EmailPalette by assigning an appro
$EmailPaletteExtraButtons and executing EmailPalette[Sequence@@$EmailPaletteExtraButtons].

Attributes for `EmailPalette`

```
{Locked, Protected, ReadProtected}
```

EmailTo

`EmailTo`

EmailTo["to","cc","bcc","subject","body"] constructs an email from the supplied data and opens it in your default email
it or edit it further. EmailTo[nb,guid] creates this email from the email content contained int he notebook nb which is
Typically in this latter case you will forward the email to your default email client using the "Send" button at the top c

Attributes for `EmailTo`

```
{Locked, Protected, ReadProtected}
```

EncryptNotebook

EncryptNotebook

EncryptNotebook["file","password"] encrypts a notebook so that it can only be opened with the indicated password. The string with no WhiteSpace characters. EncryptNotebook[nb,"password"] encrypts the notebook object nb if it has bee changes to nb since the most recent time it was saved will not be encrypted. An Encrypted notebook can be decrypted DecryptNotebook function. If you are sending an encrypted notebook to someone who does not have A WorkLife Fra ExportableDecryptingFunction[] will create a notebook that you can give them that opens a notebook with a function DecryptEncryptedNotebook that can be used by individuals who do not have access to the WorkLife FrameWork™ I notebooks that have been encrypted using EncryptNotebook.

Default options for `EncryptNotebook`

```
{DeleteEncryptedOriginal → False}
```

Attributes for `EncryptNotebook`

```
{Protected, ReadProtected}
```

EncryptNotebookDialog

EncryptNotebookDialog

EncryptNotebookDialog[] opens the EncryptNotebookDialog window.

Default options for `EncryptNotebookDialog`

```
{DeleteEncryptedOriginal → False}
```

Attributes for `EncryptNotebookDialog`

```
{Protected, ReadProtected}
```

EssayBodyCell

EssayBodyCell

EssayBodyCell[guid,style]

Default options for `EssayBodyCell`

```
{EssayBodyCellOptions → True}
```

Attributes for `EssayBodyCell`

```
{Protected, ReadProtected}
```

EssayNotesCell

```
EssayNotesCell
```

EssayNotesCell[guid]

Attributes for `EssayNotesCell`

```
{Locked, Protected, ReadProtected}
```

EssayPalette

```
EssayPalette
```

EssayPalette[] opens up the Email palette. Additional buttons can be appended to the EssayPalette palette by assigning
$EssayPaletteExtraButtons and executing EssayPalette[Sequence@@$EssayPaletteExtraButtons].

Attributes for `EssayPalette`

```
{Locked, Protected, ReadProtected}
```

ETPhoneHome

```
ETPhoneHome
```

ETPhoneHome[nb,GUID] goes to the Reiss Piece with the given GUID in the notebook nb and deletes it.

Attributes for `ETPhoneHome`

```
{Locked, Protected, ReadProtected}
```

EvaluateDefaultCodeCells

EvaluateDefaultCodeCells

EvaluateDefaultCodeCells[] evaluates the default code cells in the $CurrentDiaryNotebook.

Attributes for `EvaluateDefaultCodeCells`

```
{Locked, Protected, ReadProtected}
```

EvaluateToSelection

EvaluateToSelection

EvaluateToSelection[nb] evaluates all Input cells in the notebook nb up to and including the current selection in nb. Eva
this for the current InputNotebook. EvaluateToSelection should not be executed in a notebook with that notebook as i
EvaluateToSelection[] should not be executed within InputNotebook[]: in this case an Error message will be generate
only use EvaluateToSelection as part of the ButtonFunction of a Button.

Attributes for `EvaluateToSelection`

```
{Locked, Protected, ReadProtected}
```

EvaluateWithTracking

EvaluateWithTracking

EvaluateWithTracking[nb] evaluates the currently selected cells in the notebook nb and places a copy of them in to the c
Tracking Notebook.

Attributes for `EvaluateWithTracking`

```
{Locked, Protected, ReadProtected}
```

EvaluationPalette

EvaluationPalette

EvaluationPalette[] opens the Evaluation palette.

Attributes for `EvaluationPalette`

```
{Locked, Protected, ReadProtected}
```

EvaluationTrackingNotebook

`EvaluationTrackingNotebook`

EvaluationTrackingNotebook[] gives the current evaluation tracking notebook object if it is open. Otherwise it returns $

Attributes for `EvaluationTrackingNotebook`

```
{Locked, Protected, ReadProtected}
```

ExcludedTags

`ExcludedTags`

ExcludedTags is an option to NotebooksCellTags that is a list containing string patterns (which can be RegularExpressio versions greater than 5.0) that are excluded from the list of CellTags that are returned by NotebooksCellTags. In addit ExcludedTags may include pure functions that evaluate to True or False.

Attributes for `ExcludedTags`

```
{Locked, Protected, ReadProtected}
```

ExecuteOrganizationFlow

`ExecuteOrganizationFlow`

ExecuteOrganizationFlow[nb,"organization",{"flowTagA","flowTagB",...}] executes the OrganizationFlow given by the specified Organization if the notebook belongs to it.

ExecuteOrganizationFlow[nb,"organization",Default] executes the default OrganizationFlow given by the list of tags in OrganizationFlowTags["organization"] if the notebook nb belongs to "organization".

ExecuteOrganizationFlow[nb,"organization"] executes the OrganizationFlow corresponding the list of tags in the notebo

Attributes for `ExecuteOrganizationFlow`

```
{Locked, Protected, ReadProtected}
```

ExpandDiaryNotebook

ExpandDiaryNotebook

ExpandDiaryNotebook[] opens all subgroups of the entries in the current Diary notebook.

Default options for `ExpandDiaryNotebook`

```
{OpenDefaultCodeCell → False}
```

Attributes for `ExpandDiaryNotebook`

```
{Protected, ReadProtected}
```

ExpandNotebook

ExpandNotebook

ExpandNotebook[nb] opens all subgroups of the entries in the notebook nb.

Attributes for `ExpandNotebook`

```
{Locked, Protected, ReadProtected}
```

ExportableDecryptingFunction

ExportableDecryptingFunction

ExportableDecryptingFunction[] opens a notebook with a function DecryptEncryptedNotebook that can be used by indi
access to the WorkLife FrameWork™ Package to decrypt notebooks that have been encrypted using EncryptNoteboo

Attributes for `ExportableDecryptingFunction`

```
{Locked, Protected, ReadProtected}
```

ExportedPackageFunctions

ExportedPackageFunctions

ExportedPackageFunctions[] gives the names of the exported functions defined in the $CurrentPackageNotebook.

Attributes for `ExportedPackageFunctions`

```
{Locked, Protected, ReadProtected}
```

ExtractDateFromBackupDirectoryName

`ExtractDateFromBackupDirectoryName`

ExtractDateFromBackupDirectoryName[directoryName] extracts the date from the provided directory name of a directo
using BackupDirectory. The date is returned in your current TimeZone as set on your system. The form
ExtractDateFromBackupDirectoryName[directoryName, ToDate|AbsoluteTime] returns the date in either the form re
AbsoluteTime respectively. The form ExtractDateFromBackupDirectoryName[fileName,ToDate|AbsoluteTime,BUS
instead of "BU".

Attributes for `ExtractDateFromBackupDirectoryName`

```
{Locked, Protected, ReadProtected}
```

ExtractDateFromBackupFileName

`ExtractDateFromBackupFileName`

ExtractDateFromBackupFileName[fileName] extracts the date from the file name of a notebook that was backed up usir
date is returned in your current TimeZone as set on your system. The form ExtractDateFromBackupFileName[fileNar
returns the date in either the form returned by ToDate or AbsoluteTime respectively. The form
ExtractDateFromBackupFileName[fileName,ToDate|AbsoluteTime,BUString] uses BUString instead of "BU".

Attributes for `ExtractDateFromBackupFileName`

```
{Locked, Protected, ReadProtected}
```

FavoriteDiaries

`FavoriteDiaries`

FavoriteDiaries[] gives a list of the favorite diaries. The form of the list is {{Date, FileName}..} where Date is the date t
this list of favorites and FileName is the full path to the Diary notebook.

Attributes for `FavoriteDiaries`

```
{Locked, Protected, ReadProtected}
```

FavoriteNotebooks

`FavoriteNotebooks`

FavoriteNotebooks[] gives a list of the favorite notebooks. The form of the list is {{Date, FileName}..} where Date is th
was most recently opened and FileName is the full path to the notebook.

Attributes for `FavoriteNotebooks`

```
{Locked, Protected, ReadProtected}
```

FavoritePalettesPalette

`FavoritePalettesPalette`

FavoritePalettesPalette[] opens up the Favorite Palettes Palette.

Attributes for `FavoritePalettesPalette`

```
{Locked, Protected, ReadProtected}
```

FavoritesAndRecentOrganizer

`FavoritesAndRecentOrganizer`

FavoritesAndRecentOrganizer[] opens up a FavoritesAndRecentOrganizer.

Attributes for `FavoritesAndRecentOrganizer`

```
{Locked, Protected, ReadProtected}
```

FavoritesAndRecentPalette

`FavoritesAndRecentPalette`

FavoritesAndRecentPalette[] opens the Favorites & Recent palette.

Attributes for `FavoritesAndRecentPalette`

```
{Locked, Protected, ReadProtected}
```

FieldName

`FieldName`

FieldName[fieldName] represents a database field name to be used in constructs for DatabaseFind with DatabasePattern

Attributes for `FieldName`

```
{Locked, Protected, ReadProtected}
```

FieldTypes

`FieldTypes`

FieldTypes is an option to CreateDatabase that allows you to specify patterns that the database records must adhere to.

Attributes for `FieldTypes`

```
{Locked, Protected, ReadProtected}
```

FileNameNoDirectory

`FileNameNoDirectory`

FileNameNoDirectory[file] gives the file's name without the full path.

Attributes for `FileNameNoDirectory`

```
{Locked, Protected, ReadProtected}
```

FileSet

`FileSet`

FileSet["name"] updates $FileSets and displays the list of files in the named FileSet.

Attributes for `FileSet`

`{Locked, Protected, ReadProtected}`

FileSetNames

`FileSetNames`

FileSetNames[] gives a list of the names of FileSets that you have defined.

Attributes for `FileSetNames`

`{Locked, Protected, ReadProtected}`

FileSetsButtonData

`FileSetsButtonData`

FileSetsButtonData[] gives a list of button information for use in AssignButtonsToCustomPalette.

Attributes for `FileSetsButtonData`

`{Locked, Protected, ReadProtected}`

FindCellsBetweenDates

`FindCellsBetweenDates`

FindCellsBetweenDates[nb,tag,date1,date2] finds all cells in the notebook nb that have CellTags of the form "tag[date]"
date1 and date2. The resulting cells are displayed in a new notebook in the order that they appear in nb. The form
FindCellsBetweenDates[tag,date1,date2] performs this procedure on $CurrentDiaryNotebook.

Attributes for `FindCellsBetweenDates`

`{Locked, Protected, ReadProtected}`

FindDiaryOrNotebookOrPackage

`FindDiaryOrNotebookOrPackage`

FindDiaryOrNotebookOrPackage["string"] returns a list with information on any Diaries, Notebooks, or Packages that a
DiariesNotebooksAndPackagesDatabase.

Attributes for `FindDiaryOrNotebookOrPackage`

```
{Locked, Protected, ReadProtected}
```

FindDiaryOrNotebookOrPackageDialog

`FindDiaryOrNotebookOrPackageDialog`

FindDiaryOrNotebookOrPackageDialog[] opens a dialog that allows you to search for a Diary or Notebook by entering

Attributes for `FindDiaryOrNotebookOrPackageDialog`

```
{Locked, Protected, ReadProtected}
```

FormattingPalette

`FormattingPalette`

FormattingPalette[] opens up the formatting palette. Additional buttons can be appended to the formatting palette by ass
to $FormattingPaletteExtraButtons and executing FormattingPalette[Sequence@@$FormattingPaletteExtraButtons].

Attributes for `FormattingPalette`

```
{Locked, Protected, ReadProtected}
```

FormTemplates

`FormTemplates`

FormTemplates[] gives a list of the names of the available Form Templates.

Attributes for FormTemplates

```
{Locked, Protected, ReadProtected}
```

FormTemplatesPalette

FormTemplatesPalette

FormTemplatesPalette[] opens the Forms Template palette.

Attributes for FormTemplatesPalette

```
{Locked, Protected, ReadProtected}
```

FoundNotebooksList

FoundNotebooksList

FoundNotebooksList[dir] gives a list of notebooks contained in the directory dir and all subdirectories below it to a dept
option NotebookSearchDepth.

Default options for FoundNotebooksList

```
{Shallow → False, NotebookSearchDepth :→ $NotebookSearchDepth,
 NonHomeSubDirectories → False, IncludeBackups → True}
```

Attributes for FoundNotebooksList

```
{Protected, ReadProtected}
```

FromUniversalTimeToLocalTime

FromUniversalTimeToLocalTime

FromUniversalTimeToLocalTime[date] converts a date (assumed given in Universal Coordinated Time) into the local d
local date corresponds to the time `zone that your computer is set for.

Attributes for FromUniversalTimeToLocalTime

```
{Locked, Protected, ReadProtected}
```

FullString

FullString

FullString is an option to WebSearch that determines whether the search is done on the search full search string as a unit

Attributes for `FullString`

```
{Locked, Protected, ReadProtected}
```

FunctionDocumentationNotebook

FunctionDocumentationNotebook

FunctionDocumentationNotebook[] creates a notebook with a formatted version of the usage messages and other inform
functions in the WorkLife FrameWork™ Package. FunctionDocumentationNotebook[function] creates such a notebo
function.

Attributes for `FunctionDocumentationNotebook`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

FunctionsWithUsageMessages

FunctionsWithUsageMessages

FunctionsWithUsageMessages["Context"] gives a list of the names of functions from the given context that have usage

Attributes for `FunctionsWithUsageMessages`

```
{Locked, Protected, ReadProtected}
```

GatherCellTagData

GatherCellTagData

GatherCellTagData[head] gathers and returns a list of the CellTags in the current Diary notebook with the indicated Hea
assumed to have the form (as strings) of "head[arguments]". head should not be a string. GatherCellTagData[head,nb
the notebook nb. GatherCellTagData[head,cellStyle] and GatherCellTagData[head,cellStyle,nb] gathers only the Cell
CellStyle cellStyle (cellStyle should be a string). The various forms GatherCellTagData[tag], where tag is a string, wi

match the string pattern "tag*". For versions of Mathematica 5.1 or greater the forms GatherCellTagData[tagForm,nb GatherCellTagData[tagForm,cellStyle,nb,General] allow the argument tagForm which can be any form that is suppor System function StringCases.

Attributes for `GatherCellTagData`

```
{Locked, Protected, ReadProtected}
```

GatherComputationData

GatherComputationData

GatherComputationData[computationCellTagSuffix] takes the contents of all cells in the $CurrentDiaryNotebook that h "ComputationCell"<>computationCellTagSuffix and creates a data structure from these data for use in computations perform. For each such cell, the contribution to the data structure that is used is a list containing three elements. The f explicitly appears in the contents of the cell. The second element is a list of the CellTags of that Cell. Included in the l the DiaryDate tag, as well as possibly others. The third element is a list of the TaggingRules of that Cell.

The form GatherComputationData[{compCellTagSuffices...}] creates a data structure that is a list of the data structures cell types given through the list {compCellTagSuffices...}. The forms GatherComputationData[nb, computationCellT GatherComputationData[nb, {compCellTagSuffices...}] does this for the notebook nb.

Attributes for `GatherComputationData`

```
{Locked, Protected, ReadProtected}
```

GenerateNotebook

GenerateNotebook

GenerateNotebook is an option to DatabaseFind and other functions that specifies whether a notebook of the results shou of the function's evaluation are also returned.

Attributes for `GenerateNotebook`

```
{Locked, Protected, ReadProtected}
```

GetColorFromDialog

GetColorFromDialog

GetColorFromDialog[] returns a color directive from a color selection dialog.

Attributes for GetColorFromDialog

```
{Locked, Protected, ReadProtected}
```

GetTagDate

GetTagDate

GetTagDate[tag] returns the time associated with the tag "tag" in the currently selected cell in the current Diary noteboo
a string. GetTagDate[nb,tag] has the described function for the notebook object nb. The value returned by GetTagDat
that returned by Date[].

Attributes for GetTagDate

```
{Locked, Protected, ReadProtected}
```

GoToEndOfDiary

GoToEndOfDiary

GoToEndOfDiary[] places the insertion point at the end of $CurrentDiaryNotebook just before the Entry Toolbar Cells.

Attributes for GoToEndOfDiary

```
{Locked, Protected, ReadProtected}
```

GrabBlogName

GrabBlogName

GrabBlogName[] returns the blog name from the current blog entry in $CurrentDiaryNotebook. The cursor must be posi
this to work.

Attributes for GrabBlogName

```
{Locked, Protected, ReadProtected}
```

GrabMetaDataFromMetaDataCell

GrabMetaDataFromMetaDataCell

GrabMetaDataFromMetaDataCell[nb,guid] returns a list of the metadata for the MetaDataCell corresponding to an entry
notebook object nb.

Attributes for `GrabMetaDataFromMetaDataCell`

```
{Locked, Protected, ReadProtected}
```

GrabRSS

`GrabRSS`

GrabRSS["url"] accesses an RSS feed from the supplied URL (given as a string) and opens a formatted notebook with it
all currently specified RSS feeds (and opens and places them in a single Notebook if the GrabRSSOutput is Automati
equivalent to clicking on the "All Feeds" button on the RSS Feeds Palette.

Default options for `GrabRSS`

```
{GrabRSSOutput → Automatic}
```

Attributes for `GrabRSS`

```
{Protected, ReadProtected}
```

GrabRSSOutput

`GrabRSSOutput`

GrabRSSOutput is an option for GrabRSS. Its possible values are:

* Automatic: Opens a Notebook with the RSS feed

* Notebook: Yields a Notebook expression

* Cell: Yields a list of Cells.

Attributes for `GrabRSSOutput`

```
{Locked, Protected, ReadProtected}
```

GUID

`GUID`

GUID[] generates a 128 bit GUID. The result is returned as a base 10 integer. GUID[n] generates an n-bit GUID if n>12 generates an n-bit "GUID" even if n<128. For sufficiently small n there will be no guarantee that the GUID generatec unique.

Attributes for `GUID`

```
{Locked, Protected, ReadProtected}
```

HideAllPalettes

`HideAllPalettes`

HideAllPalettes[] hides all currently open package palettes but does not remove them from $OpenPalettes. To reopen th HideAllPalettes[False].

Attributes for `HideAllPalettes`

```
{Locked, Protected, ReadProtected}
```

HTMLSaveTemplateFiles

`HTMLSaveTemplateFiles`

HTMLSaveTemplateFiles[] gives a list of available template files in the $HTMLSaveTemplatesDirectory.

Attributes for `HTMLSaveTemplateFiles`

```
{Locked, Protected, ReadProtected}
```

HTMLSaveWithTemplate

`HTMLSaveWithTemplate`

HTMLSaveWithTemplate["TemplateFile",args___] saves a file or notebook as HTML making use of the indicated Tem arguments of HTMLSaveWithTemplate, args, are the same as those of the Mathematica function HTMLSave. The ter the values of the strings $HTMLHeadString and $HTMLBodyString so that the resulting generated HTML is placed any of the example template files for an illustration of how this should be done. The list of current Template files is gi HTMLSaveTemplateFiles[].

Attributes for `HTMLSaveWithTemplate`

```
{Locked, Protected, ReadProtected}
```

IncludeBackups

`IncludeBackups`

IncludeBackups is an option to NotebookDiscovery and FoundNotebooksList that determines whether backup notebook
for NotebookDiscovery is False and for FoundNotebooksList is True.

Attributes for `IncludeBackups`

```
{Locked, Protected, ReadProtected}
```

IncludeCellCreationHistory

`IncludeCellCreationHistory`

IncludeCellCreationHistory is an option to NotebookOpeningHistoryGraphicCell the determines whether data on when
created (via their DiaryDate tags) are included in the graphic.

Attributes for `IncludeCellCreationHistory`

```
{Locked, Protected, ReadProtected}
```

IncludeDate

`IncludeDate`

IncludeDate is an option for TagCell and other functions that determines whether time tag information is included.

Attributes for `IncludeDate`

```
{Locked, Protected, ReadProtected}
```

IncludeDayOfTheWeek

`IncludeDayOfTheWeek`

IncludeDayOfTheWeek is an option for CalendarDate that determines whether the day of the week will be printed in the
True.

Attributes for `IncludeDayOfTheWeek`

```
{Locked, Protected, ReadProtected}
```

IncludedTags

`IncludedTags`

IncludedTags is an option to NotebooksCellTags that is a list containing string patterns (which can be RegularExpressio
greater than 5.0) that specify those tags that will be included in the list of CellTags that are returned by NotebooksCel
even if a tag satisfies this criterion it may not be returned if it is excluded by the ExcludedTags option.

Attributes for `IncludedTags`

```
{Locked, Protected, ReadProtected}
```

IncludeImageLocationInCellTag

`IncludeImageLocationInCellTag`

IncludeImageLocationInCellTag is an option for PlaceImage. Its default value is Automatic. In this case the imagefile s
CellTags of the cell that is created if it comes from a URL (thus keeping information on its location in your file syster
from a hyperlink). Other possible values are True and False.

Attributes for `IncludeImageLocationInCellTag`

```
{Locked, Protected, ReadProtected}
```

IncludeNotebookVersionOfBlogPost

`IncludeNotebookVersionOfBlogPost`

IncludeNotebookVersionOfBlogPost is an option for PublishBlogEntry that determines whether a notebook version of tl
in the same directory as the blog post and will be linked to from the post with a "Notebook" link next to the Permalink

Attributes for `IncludeNotebookVersionOfBlogPost`

```
{Locked, Protected, ReadProtected}
```

IncludeSeconds

IncludeSeconds

IncludeSeconds is an option for CalendarDate that determines whether the number of seconds will be printed in the outp
True.

Attributes for `IncludeSeconds`

```
{Locked, Protected, ReadProtected}
```

IncludeTime

IncludeTime

IncludeTime is an option for CalendarDate that determines whether the time of day will be printed in the output. Its defa

Attributes for `IncludeTime`

```
{Locked, Protected, ReadProtected}
```

IndentCell

IndentCell

IndentCell[nb] indents the text of the selected cell in the notebook nb by the amount $IndentCellDefault. IndentCell[] do
InputNotebook[].

Attributes for `IndentCell`

```
{Locked, Protected, ReadProtected}
```

IndexDatabase

IndexDatabase

IndexDatabase[name,i] indexes the ith field of the database given by name.

Attributes for `IndexDatabase`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

Indexed

Indexed

Indexed[name,i] is True if the ith field of the database given by name has been indexed and False otherwise. If there is n

Attributes for `Indexed`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

InputCellsFromEvaluationTrackingNotebook

InputCellsFromEvaluationTrackingNotebook

InputCellsFromEvaluationTrackingNotebook[] creates a notebook that contains only the input cells from the current Eva

Attributes for `InputCellsFromEvaluationTrackingNotebook`

```
{Locked, Protected, ReadProtected}
```

InputNotebookDirectory

InputNotebookDirectory

InputNotebookDirectory[] gives the directory where InputNotebook[] is located.

Attributes for `InputNotebookDirectory`

```
{Locked, Protected, ReadProtected}
```

InsertPageBreak

InsertPageBreak

InsertPageBreak[nb,After|Before] inserts a page break after or before the selected cell or the cell where the insertion poi
insertion point is between cells then the page break will be either at the top of the following cell or the bottom of the p

InsertPageBreak returns the notebook selection of the cell that was marked for the page break. It returns $Failed if it v
page break (for example if the insertion point was at the end of the notebook and the page break was requested to be A

Attributes for `InsertPageBreak`

```
{Locked, Protected, ReadProtected}
```

InvertPalettes

InvertPalettes

InvertPalettes[] inverts the color schemes of the palettes. To have the change take effect for all open palettes evaluate Re

Attributes for `InvertPalettes`

```
{Locked, Protected, ReadProtected}
```

LeaveOpen

LeaveOpen

LeaveOpen is an option for CloseAllPalettes which specifies those palettes which should not be closed. Its default value
{"$WorkLifeToolsPaletteNotebook"}.

Attributes for `LeaveOpen`

```
{Locked, Protected, ReadProtected}
```

LoadDatabase

LoadDatabase

LoadDatabase[name,file] loads a database residing in the file "file" and assigns it the name "name." LoadDatabase[nam
the name "name" if it exists and is listed in $Databases. "name" must be a symbol without a value.

Default options for `LoadDatabase`

```
{IndexedFields → All, Directory → Automatic}
```

Attributes for `LoadDatabase`

```
{HoldFirst, Protected, ReadProtected}
```

LoadedDatabaseQ

LoadedDatabaseQ

LoadedDatabaseQ[db] is True if the database db has been loaded.

Attributes for `LoadedDatabaseQ`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

LoadMathematicaUsageDatabase

LoadMathematicaUsageDatabase

LoadMathematicaUsageDatabase[] loads the MathematicaUsageDatabase.

Attributes for `LoadMathematicaUsageDatabase`

```
{Locked, Protected, ReadProtected}
```

LoadPlugins

LoadPlugins

LoadPlugins[] loads all of the available plug-ins. The list of available plug-in contexts is given by PluginContexts[]. Lo
loads the plug-in with the specified context.

Attributes for `LoadPlugins`

```
{Locked, Protected, ReadProtected}
```

LoadWorkLifeSkin

LoadWorkLifeSkin

LoadWorkLifeSkin[skin] loads the specified WorkLife skin if it exists. A list of WorkLife skins is given by WorkLifeS
WorkLife skin, use the SkinParametersTool by executing SkinParametersTool[]. LoadWorkLifeSkin[] or LoadWorkI
the package default skin.

Attributes for `LoadWorkLifeSkin`

```
{Locked, Protected, ReadProtected}
```

LockCell

`LockCell`

LockCell[] locks the specific cells that are chosen in the current Diary notebook. LockCell[nb] locks the specific cells th
notebook nb.

Attributes for `LockCell`

```
{Locked, Protected, ReadProtected}
```

LockCells

`LockCells`

LockCells[] locks all of the cells of the types listed in $LockingCellStyles in the current Diary notebook. Any new entri
until LockCells[] is executed again. LockCells[All] locks all cells in the current Diary notebook.

Attributes for `LockCells`

```
{Locked, Protected, ReadProtected}
```

MailBoxType

`MailBoxType`

MailBoxType is an option to EmailHeaders and similar functions.

Attributes for `MailBoxType`

```
{Locked, Protected, ReadProtected}
```

MailBoxTypes

`MailBoxTypes`

MailBoxTypes[] returns a list of the permissable mailbox types.

Attributes for `MailBoxTypes`

```
{Locked, Protected, ReadProtected}
```

MakeCalendar

`MakeCalendar`

MakeCalendar[Function] creates a calendar dialog that executes Function.

Default options for `MakeCalendar`

```
{CalendarTitleBarBackground :→ $CalendarTitleBarBackground,
 CalendarDaysOfWeekButtonBackground :→ $CalendarDaysOfWeekButtonBackground,
 CalendarDatesButtonBackground :→ $CalendarDatesButtonBackground,
 CalendarCurrentDayButtonBackground :→ $CalendarCurrentDayButtonBackground,
 CalendarNotebookBackground :→ $CalendarNotebookBackground}
```

Attributes for `MakeCalendar`

```
{HoldFirst, Protected, ReadProtected}
```

MakeCurrent

`MakeCurrent`

MakeCurrent[nb] makes the notebook object nb the Current Diary if it is a Diary.

Attributes for `MakeCurrent`

```
{Locked, Protected, ReadProtected}
```

MakeEmailForm

`MakeEmailForm`

MakeEmailForm[nb] creates an email form in the notebook nb.

Attributes for `MakeEmailForm`

```
{Locked, Protected, ReadProtected}
```

MakeEssay

`MakeEssay`

MakeEssay[nb] creates an Essay in the notebook nb. MakeEssay[] creates an essay in the current Diary notebook.

Attributes for `MakeEssay`

```
{Locked, Protected, ReadProtected}
```

MakeGeneralPalette

`MakeGeneralPalette`

MakeGeneralPalette[{buttonCellData..}] can be used to make a custom palette. When MakeGeneralPalette is executed i
the AllOpenNotebooksPalette if it is open.

Attributes for `MakeGeneralPalette`

```
{Locked, Protected, ReadProtected}
```

MakeStandardDiaryDirectory

`MakeStandardDiaryDirectory`

MakeStandardDiaryDirectory[] creates a standard Diary directory named "Diaries" as a an appropriate subdirectory of $
in Windows and Mac OS X: for Windows it is in ToFileName[{$HomeDirectory,"My Documents","Diaries"}], and f
ToFileName[{$HomeDirectory,"Documents","Diaries"}]) and sets it as the default Diary directory. By Default a file
placed in this directory. MakeStandardDiaryDirectory["filename"] will place the file "filename.nb".

Attributes for `MakeStandardDiaryDirectory`

```
{Locked, Protected, ReadProtected}
```

MarkDone

MarkDone

MarkDone[] marks the currently selected cell in the current Diary notebook as a "Done" item if it was previously marke
MarkDone[nb] marks a ToDo in the notebook nb as Done.

Attributes for `MarkDone`

```
{Locked, Protected, ReadProtected}
```

MarkToDo

MarkToDo

MarkToDo[] marks the currently selected cell in the current Diary notebook as a "ToDo" item.

Attributes for `MarkToDo`

```
{Locked, Protected, ReadProtected}
```

MathematicaUsageDatabaseArchiveFiles

MathematicaUsageDatabaseArchiveFiles

MathematicaUsageDatabaseArchiveFiles[] gives the full paths to the MathematicaUsageDatabase archive files.

Attributes for `MathematicaUsageDatabaseArchiveFiles`

```
{Locked, Protected, ReadProtected}
```

MathematicaUsageDatabaseItemTags

MathematicaUsageDatabaseItemTags

MathematicaUsageDatabaseItemTags[] gives the list of the possible item tags for the "ItemTag" field name of the Mathe

Attributes for `MathematicaUsageDatabaseItemTags`

```
{Locked, Protected, ReadProtected}
```

ModifyDatabaseRecord

`ModifyDatabaseRecord`

ModifyDatabaseRecord[name,originalRecord,replacementRecord] replaces the record originalRecord with replacementl database. If there is more than one copy of originalRecord in the database then all of them are replaced with replacem

Attributes for `ModifyDatabaseRecord`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

MostRecentOpeningDate

`MostRecentOpeningDate`

MostRecentOpeningDate[nb] gives the most recent date that the NotebookObject nb was opened by the WorkLife Fram MostRecentOpeningDate["file"] gives the most recent date that the Notebook file "file" was opened by the WorkLife given in the Local Time of your system settings.

Attributes for `MostRecentOpeningDate`

```
{Locked, Protected, ReadProtected}
```

MoveDingbatIntoCell

`MoveDingbatIntoCell`

MoveDingbatIntoCell[nb] --if the currently selected cell in the notebook nb has a cell dingbat, MoveDingbatIntoCell mc content of the cell. This is useful, for example, for when one is saving a notebook as a web page via HTMLSave sinc reproduced in the html version of the notebook.

Attributes for `MoveDingbatIntoCell`

```
{Locked, Protected, ReadProtected}
```

NewDiaryNotebook

NewDiaryNotebook

NewDiaryNotebook[notebookName] creates a new Diary notebook with the name "notebookName.nb" and saves it into
If the option NewDirectory is set to True (the default value is False), then the Diary is created within the directory "nc
$DefaultDiaryDirectory. NewDiaryNotebook[{notebookNames...}] creates multiple Diaries within $DefaultDiaryDir
NewDiaryNotebook/@{notebookNames...} will create several new Diaries within $DefaultDiaryDirectory.

Default options for `NewDiaryNotebook`

```
{NewDirectory → False, DiaryDirectoryFunction → (#1 &)}
```

Attributes for `NewDiaryNotebook`

```
{Protected, ReadProtected}
```

NewDiaryNotebookDialog

NewDiaryNotebookDialog

NewDiaryNotebookDialog[] opens the new Diary notebook dialog.

Default options for `NewDiaryNotebookDialog`

```
{CurrentDirectoryOnly → False}
```

Attributes for `NewDiaryNotebookDialog`

```
{Protected, ReadProtected}
```

NewDirectory

NewDirectory

NewDirectory is an option to NewDiaryNotebook that determines whether a new Diary notebook will reside in its own c
$CurrentDiaryNotebookDirectory or whether it will be placed in the $CurrentDiaryNotebookDirectory.

Attributes for `NewDirectory`

```
{Locked, Protected, ReadProtected}
```

NewFileSet

NewFileSet

NewFileSet["name"] creates a new FileSet with the given name.

Attributes for `NewFileSet`

```
{Locked, Protected, ReadProtected}
```

NewNotebook

NewNotebook

NewNotebook[name] creates a new notebook with the name name.nb in the current Notebooks directory. The current va given by DiaryNotebooksDirectory[]. NewNotebook[] and NewNotebook[Default] open a dialog box to enter the nan the latter case the new notebook's default options are given by $NewNotebookDefaultOptions. When using the dialog value of the function is not the new notebook's NotebookObject.

Attributes for `NewNotebook`

```
{Locked, Protected, ReadProtected}
```

NewPackage

NewPackage

NewPackage[name] creates a new package with the name name.nb and the associated name.m in the Packages directory

Default options for `NewPackage`

```
{Diary`Diary`Private`PackageFile → Automatic}
```

Attributes for `NewPackage`

```
{Protected, ReadProtected}
```

NewRecords

NewRecords

NewRecords[name] gives a list of the recently added records to the database given by name.

Attributes for `NewRecords`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

NewScratchNotebook

NewScratchNotebook

NewScratchNotebook[name] creates a new scratch notebook with the name name*v*.nb in the current Notebooks's Scratcl
v in name*v* is automatically added to the name incremented by one above the current maximum value for files of the f
Scratch Notebook Directory. The current value of this directory is given by DiaryScratchNotebooksDirectory[]. New
NewScratchNotebook[Default] open a dialog box to enter the name of the new scratch notebook. In the latter case the
default options are given by $NewScratchNotebookDefaultOptions. When using the dialog box method the returned
the new notebook's NotebookObject.

Attributes for `NewScratchNotebook`

```
{Locked, Protected, ReadProtected}
```

NonHomeSubDirectories

NonHomeSubDirectories

NonHomeSubDirectories is an option to NotebookDiscovery that specifies whether NotebookDiscovery will be permitte
directory is not on a path below the users $HomeDirectory. Its default value is False.

Attributes for `NonHomeSubDirectories`

```
{Locked, Protected, ReadProtected}
```

NotebookBackups

NotebookBackups

NotebookBackups[nb] gives a list of backups that have been made for the NotebookObject nb. It is presented as a list wl
The first member of the pair is the date in the current computer's time zone when the backup was made. The second r
name of the corresponding backup. NotebookBackups[nb,TabularReport] produces a formatted table with buttons tha
notebook. NotebookBackups[file] and NotebookBackups[file,TabularReport] have a similar action with regard to bac
as the file "file" on disk.

Attributes for NotebookBackups

```
{Locked, Protected, ReadProtected}
```

NotebookCreationDate

NotebookCreationDate

NotebookCreationDate[nb] gives the date that the notebook nb was created if it was created by this package. If not, then was opened by this package. If neither of these apply, it gives None. NotebookCreationDate["file"] gives the creation location specified by "file". Date is given in the Local Time of your system settings.

Attributes for NotebookCreationDate

```
{Locked, Protected, ReadProtected}
```

NotebookData

NotebookData

NotebookData[nb] returns a list {name,filePath,guid,notebookType,organization} that gives the name, file path, GUID, organizations for the notebook nb. Possible types are "Diary", "Notebook", "Package", or "SlideShow".

Attributes for NotebookData

```
{Locked, Protected, ReadProtected}
```

NotebookDiscovery

NotebookDiscovery

NotebookDiscovery[dir] starts at the directory dir and looks at all directories that are up to the value of its NotebookSea find Mathematica notebooks (.nb files). The found files are added to the DiariesNotebooksAndPackagesDatabase so FrameWork will know about them.

Default options for NotebookDiscovery

```
{Shallow → True, NotebookSearchDepth :→ $NotebookSearchDepth,
 AddRecordToDiariesNotebooksAndPackagesDatabase → True,
 AddGUIDToNotebookAndSave → False, NonHomeSubDirectories → False, IncludeBackups →
```

Attributes for `NotebookDiscovery`

```
{Protected, ReadProtected}
```

NotebookFromEssay

NotebookFromEssay

NotebookFromEssay[] creates a new notebook containing the current essay. The current essay is the one that is selected
items selected or has the insertion point within it) in the current input notebook.

Attributes for `NotebookFromEssay`

```
{Locked, Protected, ReadProtected}
```

NotebookGUID

NotebookGUID

NotebookGUID[nb] gives the GUID for the notebook object nb. If nb does not have a GUID then NotebookGUID retur
NotebookGUID[] gives the GUID of the notebook that it is evaluated in.

Attributes for `NotebookGUID`

```
{Locked, Protected, ReadProtected}
```

NotebookNBQ

NotebookNBQ

NotebookNBQ["file"] is True if the indicated file exists and its file name ends in ".nb".

Attributes for `NotebookNBQ`

```
{Locked, Protected, ReadProtected}
```

NotebookOpeningHistory

NotebookOpeningHistory

NotebookOpeningHistory[nb] gives a sorted list of the dates that the notebook nb was opened by the WorkLife FrameW
the Local Time of your system settings.

Attributes for `NotebookOpeningHistory`

```
{Locked, Protected, ReadProtected}
```

NotebookOpeningHistoryGraphicCell

`NotebookOpeningHistoryGraphicCell`

NotebookOpeningHistoryGraphicCell[nb] prints a cell with a graphic representing the times when the notebook nb was
NotebookOpeningHistoryGraphicCell[nb,{date1,date2}] prints the cell for those times it was opened between date1 a
date2 should be a list of the form returned by Date[]. However only the year, month, and day are used (and so a form
acceptable). With the default option setting CellPrint→True a cell is printed with the graphic. With the option CellPri
expression itself is returned. Dates are given in the Local Time of your system settings.

Default options for `NotebookOpeningHistoryGraphicCell`

```
{CellPrint → True, PlotStyle → {RGBColor[0.7, 0, 0], AbsoluteThickness[2]},
 IncludeCellCreationHistory → True,
 CellCreationHistoryPointStyle → {AbsolutePointSize[4], RGBColor[0, 0.2, 0.9]}}
```

Attributes for `NotebookOpeningHistoryGraphicCell`

```
{Protected, ReadProtected}
```

NotebookOpeningHistoryReport

`NotebookOpeningHistoryReport`

NotebookOpeningHistoryReport[nb] gives a list of properties of the history of the notebook nb.

Attributes for `NotebookOpeningHistoryReport`

```
{Locked, Protected, ReadProtected}
```

NotebookOpenQ

`NotebookOpenQ`

NotebookOpenQ[nb] tests whether the notebook object nb corresponds to a currently open notebook. If nb does not corr
NotebookObject then NotebookOpenQ[nb] returns False and generates a warning message.

Attributes for `NotebookOpenQ`

```
{Locked, Protected, ReadProtected}
```

NotebooksCellTags

NotebooksCellTags

NotebooksCellTags[nb] gives a list of all of the CellTags in the notebook "nb" except for those excluded by the value of
The tags that are returned can be specified through the IncludedTags option.

Default options for `NotebooksCellTags`

```
{ExcludedTags → {*[*], After*, CellGUID*, DefaultCodeCell, DiaryDate*, Done*,
   DropReissPiece, DueDate*, EmailBodyCell*, EmailBottomCell*, EmailCell*, EmailNo
   EmailToolbarCell*, EntryToolbarCell, EphemeralGUID*, EssayBodyCell*, EssayBotto
   EssayCell*, EssayNotesCell*, EssayToolbarCell*, GUIDTag*, NotebookBackups,
   SaveBackupToolbarCell*, SubjectCell*, ToCell*, ToDo*, TopToolbarCell, CreatedOn
   ConvertedOn, Info*, ToDoPriority*, Archived*, ArchiveDiary, ComputationCellOper
   ComputationCellOperatorName, ComputationCellOperatorTemplate, IntegerQ[ToExpre
 IncludedTags → {*}, NotebooksCellTagsOutput → Automatic, Method → Automatic}
```

Attributes for `NotebooksCellTags`

```
{Protected, ReadProtected}
```

NotebooksCellTagsOutput

NotebooksCellTagsOutput

NotebooksCellTagsOutput is an option to NotebooksCellTags that determines the sort of output that the function returns
NotebooksCellTagsOutput→Automatic, for which NotebooksCellTags returns a list of the notebook's CellTags gover
ExcludedTags and IncludedTags.

Attributes for `NotebooksCellTagsOutput`

```
{Locked, Protected, ReadProtected}
```

NotebookSearchDepth

NotebookSearchDepth

NotebookSearchDepth is an option to NotebookDiscovery that specifies the number of directories below the specified o
NotebookDiscovery[dir] looks to find Mathematica notebooks. Its default value is the value of $NotebookSearchDept

Attributes for `NotebookSearchDepth`

```
{Locked, Protected, ReadProtected}
```

NotebooksInDirectory

NotebooksInDirectory

NotebooksInDirectory[dir] gives a list of notebook files that are in the directory dir.

Attributes for `NotebooksInDirectory`

```
{Locked, Protected, ReadProtected}
```

NotebooksPalette

NotebooksPalette

NotebooksPalette[] opens the Notebooks palette.

Attributes for `NotebooksPalette`

```
{Locked, Protected, ReadProtected}
```

NotebookStylesPalette

NotebookStylesPalette

NotebookStylesPalette[] opens up the Notebook Styles palette.

Attributes for NotebookStylesPalette

```
{Locked, Protected, ReadProtected}
```

NotebookTaggingRules

NotebookTaggingRules

NotebookTaggingRules[nb] gives the list of the TaggingRules (corresponding to the right hand side of the notebook's T
the notebook object nb. NotebookTaggingRules[] gives the TaggingRules for the $CurrentDiaryNotebook if it is oper
NotebookTaggingRules[nb,stringPattern] gives those TaggingRules that are strings matching the given string pattern.
NotebookTaggingRules[stringPattern] gives this for $CurrentDiaryNotebook.

Attributes for NotebookTaggingRules

```
{Locked, Protected, ReadProtected}
```

NotebookType

NotebookType

NotebookType[nb] gives the notebook type of the NotebookObject nb. The possible notebook types are given by $Notel
string.

Attributes for NotebookType

```
{Locked, Protected, ReadProtected}
```

NotebookWLOQ

NotebookWLOQ

NotebookWLOQ[nb] is True if the notebook nb is a Notebook as the term is used in the WorkLife FrameWork™ packa
NotebookWLOQ["file"] is True if the indicated file exists and is a Notebook as the term is used in the WorkLife Fran

Attributes for NotebookWLOQ

```
{Locked, Protected, ReadProtected}
```

NumberOfDatabaseFields

`NumberOfDatabaseFields`

NumberOfDatabaseFields[name] gives the number of fields in the database. The field names can be obtained from Data

Attributes for `NumberOfDatabaseFields`

```
{Locked, Protected, ReadProtected}
```

NumberOfDatabaseRecords

`NumberOfDatabaseRecords`

NumberOfDatabaseRecords[name] gives the number of records in the database.

Attributes for `NumberOfDatabaseRecords`

```
{Locked, Protected, ReadProtected}
```

NumberOfWorkLifeFunctions

`NumberOfWorkLifeFunctions`

NumberOfWorkLifeFunctions[] gives the number of functions and parameters exported by this package. Only those fun
have usage messages should be considered to be user-friendly. These functions and parameters are listed in the paran
$FunctionsWithUsageMessages. And the number of such functions and parameters is given by
NumberOfWorkLifeFunctionsWithUsageMessages[]. Other functions and parameters in this package are generally in
should only be used with care if at all as they are not documented, and they may perform unexpected things.

Attributes for `NumberOfWorkLifeFunctions`

```
{Locked, Protected, ReadProtected}
```

NumberOfWorkLifeFunctionsWithUsageMessages

`NumberOfWorkLifeFunctionsWithUsageMessages`

NumberOfWorkLifeFunctionsWithUsageMessages[] gives the number of functions and parameters exported by this pac
be user-friendly. Only those functions and parameters that have usage messages should be considered to be user-frien

parameters are listed in the parameter $FunctionsWithUsageMessages. Other functions and parameters in this packag
for internal use and should only be used with care if at all as they are not documented, and they may perform unexpec

Attributes for `NumberOfWorkLifeFunctionsWithUsageMessages`

```
{Locked, Protected, ReadProtected}
```

OpenAllPalettes

`OpenAllPalettes`

OpenAllPalettes[] opens all palettes associated with this package, if possible. These are the notebooks whose names are
Names["Diary`Diary`$*PaletteNotebook"].

Attributes for `OpenAllPalettes`

```
{Locked, Protected, ReadProtected}
```

OpenArchive

`OpenArchive`

OpenArchive[] opens the most recent archive for the current Diary, if that archive exists.

Attributes for `OpenArchive`

```
{Locked, Protected, ReadProtected}
```

OpenCurrentPackageNotebook

`OpenCurrentPackageNotebook`

OpenCurrentPackageNotebook[] opens the most recently opened package notebook file given by $CurrentPackageNotel

Attributes for `OpenCurrentPackageNotebook`

```
{Locked, Protected, ReadProtected}
```

OpenDefaultCodeCell

OpenDefaultCodeCell

OpenDefaultCodeCell is an option for ExpandDiaryNotebook and UnlockCells that determines whether or not the Defau
opened if they are closed. It's default value is OpenDefaultCodeCell→False.

Attributes for `OpenDefaultCodeCell`

```
{Locked, Protected, ReadProtected}
```

OpenDefaultPaletteSet

OpenDefaultPaletteSet

OpenDefaultPaletteSet[] resets the open Palettes to be the user-specified default palette set. To set a default palette set us
SetDefaultPaletteSet.

Attributes for `OpenDefaultPaletteSet`

```
{Locked, Protected, ReadProtected}
```

OpenDiaryNotebook

OpenDiaryNotebook

OpenDiaryNotebook[] opens the Diary notebook file given by $CurrentDiaryNotebookFile. OpenDiaryNotebook[opts]
options "opts."

Attributes for `OpenDiaryNotebook`

```
{Locked, Protected, ReadProtected}
```

OpenDiaryNotebookFile

OpenDiaryNotebookFile

OpenDiaryNotebookFile[file] opens the file if it is a Diary.

Attributes for `OpenDiaryNotebookFile`

```
{Locked, Protected, ReadProtected}
```

OpenEmailClient

`OpenEmailClient`

OpenEmailClient[] opens the default email client on your computer with a fresh email.

Attributes for `OpenEmailClient`

```
{Locked, Protected, ReadProtected}
```

OpenFileOrDirectory

`OpenFileOrDirectory`

OpenFileOrDirectory[fileOrDirectoryName] opens the file or directory given by the string fileOrDirectoryName. If file(
notebook or other Mathematica object, it will open in the appropriate application for its file type. If fileOrDirectoryN
that application will be launched. If fileOrDirectoryName is a web URL then that URL will be opened in your default
function OpenURLInBrowser is the more direct way to do this in the case of a URL rather than a file on your local fil

Attributes for `OpenFileOrDirectory`

```
{Locked, Protected, ReadProtected}
```

OpenFileSet

`OpenFileSet`

OpenFileSet["name"] opens the files in $FileSets corresponding to the FileSet with the name "name". OpenFileSet[{file
files in their default applications. The files must all be strings representing full file paths. Those that begin in http:// o
your default web browser. To add files to or delete files from $FileSets use the functions AddFileTo$FileSetsDialog[]
DeleteFileFrom$FileSetsDialog[].

Attributes for `OpenFileSet`

```
{Locked, Protected, ReadProtected}
```

OpenFileSetDialog

OpenFileSetDialog

OpenFileSetDialog[] opens a dialog that allows you to open a FileSet.

Attributes for `OpenFileSetDialog`

`{Locked, Protected, ReadProtected}`

OpenNotebookWithGUID

OpenNotebookWithGUID

OpenNotebookWithGUID[guid] finds and opens the notebook that has the NotebookGUID given by guid if it is contain
DiariesNotebooksAndPackagesDatabase.

Attributes for `OpenNotebookWithGUID`

`{Locked, Protected, ReadProtected}`

OpenURLInBrowser

OpenURLInBrowser

OpenURLInBrowser[url] opens the specified url in your default web browser. The url must be a string that begins in htt

Attributes for `OpenURLInBrowser`

`{Locked, Protected, ReadProtected}`

Organization

Organization

Organization[nb] gives the full organization data for the notebook object nb. Organization["organization"] gives the ful
Organization "organization".

Attributes for `Organization`

```
{Locked, Protected, ReadProtected}
```

OrganizationDiaries

`OrganizationDiaries`

OrganizationDiaries["organization"] gives a list of all of the Diaries that are part of the Organization "organization". Ea
the form of a length two list. The first item in each list is the Diary's name, and the second item is the full path to that

Attributes for `OrganizationDiaries`

```
{Locked, Protected, ReadProtected}
```

OrganizationFlowTags

`OrganizationFlowTags`

OrganizationFlowTags["organization"] gives the list of tags associated with the organization rules for the Organization '
of the tags is the same as the order of the corresponding rules in OrganizationRules["organization"]. OrganizationFlov
tags associated with the OrganizationFlow for the notebook nb if it is a member of an Organization.

Attributes for `OrganizationFlowTags`

```
{Locked, Protected, ReadProtected}
```

OrganizationNotebooks

`OrganizationNotebooks`

OrganizationNotebooks["organization"] gives a list of all of the Notebooks that are part of the Organization "organizatic
list is in the form of a length two list. The first item in each list is the Notebook's name, and the second item is the ful

Attributes for `OrganizationNotebooks`

```
{Locked, Protected, ReadProtected}
```

OrganizationQ

OrganizationQ

OrganizationQ["organization"] is True if "organization" is an Organization.

Attributes for `OrganizationQ`

```
{Locked, Protected, ReadProtected}
```

OrganizationRules

OrganizationRules

OrganizationRules["organization"] gives the rules associated with the Organization "organization".

Attributes for `OrganizationRules`

```
{Locked, Protected, ReadProtected}
```

OrganizationsPalette

OrganizationsPalette

OrganizationsPalette[] opens the Organizations Palette.

Attributes for `OrganizationsPalette`

```
{Locked, Protected, ReadProtected}
```

OriginatingTimeZone

OriginatingTimeZone

OriginatingTimeZone is an option for CalendarDate and DateStringFromTag that specifies which time zone the date sup
from. The default value is OriginatingTimeZone→0.

Attributes for OriginatingTimeZone

```
{Locked, Protected, ReadProtected}
```

OtherFilesPalette

OtherFilesPalette

OtherFilesPalette[] opens the OtherFiles Palette.

Attributes for OtherFilesPalette

```
{Locked, Protected, ReadProtected}
```

OtherTags

OtherTags

OtherTags is an option to DropReissPiece that specifies a list of tags to add to the CellTags of the dropped Reiss piece.

Attributes for OtherTags

```
{Locked, Protected, ReadProtected}
```

OtherToolsButtonData

OtherToolsButtonData

OtherToolsButtonData[] gives a list of button information for use in AssignButtonsToCustomPalette.

Attributes for OtherToolsButtonData

```
{Locked, Protected, ReadProtected}
```

OutdentCell

OutdentCell

OutdentCell[] outdents the text of the selected cell in the notebook nb by the amount $IndentCellDefault. OutdentCell[]
InputNotebook[].

Attributes for `OutdentCell`

```
{Locked, Protected, ReadProtected}
```

PackageFunctionCategories

`PackageFunctionCategories`

PackageFunctionCategories[] gives the function categories in the $CurrentPackageNotebook.

Attributes for `PackageFunctionCategories`

```
{Locked, Protected, ReadProtected}
```

PackageProgrammingPalette

`PackageProgrammingPalette`

PackageProgrammingPalette[] opens the Package Programming Palette for the $CurrentPackageNotebook if it is define

Attributes for `PackageProgrammingPalette`

```
{Locked, Protected, ReadProtected}
```

PackageQ

`PackageQ`

PackageQ[nb] is True if the notebook nb is a Package as the term is used in the WorkLife FrameWork™ package. Pack
indicated file exists and is a Package as the term is used in the WorkLife FrameWork™ package.

Attributes for `PackageQ`

```
{Locked, Protected, ReadProtected}
```

PackagesDirectory

PackagesDirectory

PackagesDirectory[] gives the directory where packages associated with the Diary are stored.

Attributes for `PackagesDirectory`

```
{Locked, Protected, ReadProtected}
```

PackagesPalette

PackagesPalette

PackagesPalette[] opens the Packages palette.

Attributes for `PackagesPalette`

```
{Locked, Protected, ReadProtected}
```

PackageTemplate

PackageTemplate

PackageTemplate is an option to NewPackage. With PackageTemplate→Automatic the standard package template is use
PackageTemplate→Plugin the Plugin template is used.

Attributes for `PackageTemplate`

```
{Locked, Protected, ReadProtected}
```

PaletteButtonCell

PaletteButtonCell

PaletteButtonCell[{"title",function,{textStyle},{buttonOptions}},"cellStyle",cellOptions] or
PaletteButtonCell[{{"title",function,{textStyle},{buttonOptions}}...},"cellStyle",cellOptions] can be used to create a
first form creates a single button, and the second form creates multiple buttons in a row. The "title" is displayed on the
CellStyle will be "cellStyle". The button's ButtonFunction, function, must be a pure function. textStyle gives the text

text, and buttonOptions gives further options for the button. Typical values for {textStyle} and {buttonOptions} are $
and $GeneralButtonButtonOptions respectively.

Default options for `PaletteButtonCell`

```
{ButtonDocumentation → Automatic, TheButtonBoxNoMessages → False}
```

Attributes for `PaletteButtonCell`

```
{Protected, ReadProtected}
```

PaletteWindowTitle

`PaletteWindowTitle`

PaletteWindowTitle[palette] gives the window title of the palette. The palette does not have to be open. The argument t
either be either the palette function (for example, WorkLifeToolsPalette) or the name of the palette function (for exan
"WorkLifeToolsPalette").

Attributes for `PaletteWindowTitle`

```
{Locked, Protected, ReadProtected}
```

PasswordProtectButtonCell

`PasswordProtectButtonCell`

PasswordProtectButtonCell[] is gives a Cell Expression for a Cell with a password protect button.

Attributes for `PasswordProtectButtonCell`

```
{Locked, Protected, ReadProtected}
```

PastDueToDos

`PastDueToDos`

PastDueToDos[] opens a notebook with those ToDos (of those that have been assigned a DueDate) which are past their

Default options for `PastDueToDos`

```
{ShowEmptyNotebook → True}
```

Attributes for PastDueToDos

```
{Protected, ReadProtected}
```

PasteColorFromDialog

PasteColorFromDialog

PasteColorFromDialog[nb] pastes a color directive from a color selection dialog at the current insertion point in the note
PasteColorFromDialog[] pastes into InputNotebook[].

Attributes for PasteColorFromDialog

```
{Locked, Protected, ReadProtected}
```

PasteDate

PasteDate

PasteDate[nb] pastes the indicated date into the notebook nb at its current selection location. PasteDate[] does this in the
PasteDate is useful for supplying arguments to functions such as FindCellsBetweenDates.

Attributes for PasteDate

```
{Locked, Protected, ReadProtected}
```

PasteFormTemplateIntoNotebook

PasteFormTemplateIntoNotebook

PasteFormTemplateIntoNotebook["name"] pastes the Form Template with name "name." into the current InputNoteboo
available Form Templates execute the function FormTemplates[].

Attributes for PasteFormTemplateIntoNotebook

```
{Locked, Protected, ReadProtected}
```

PasteFunctionTemplate

PasteFunctionTemplate

PasteFunctionTemplate[nb] creates a function template at the current selection in the notebook nb.

Attributes for `PasteFunctionTemplate`

```
{Locked, Protected, ReadProtected}
```

PlaceImage

`PlaceImage`

PlaceImage[nb,imageFile] places the image from the imageFile into the notebook nb. It is placed just below the currentl

Default options for `PlaceImage`

```
{ScaleImage → Automatic, CenterGraphics → True,
 BitMapImage → True, IncludeImageLocationInCellTag → Automatic}
```

Attributes for `PlaceImage`

```
{Protected, ReadProtected}
```

PluginContexts

`PluginContexts`

PluginContexts[] gives the list of available plug-in contexts.

Attributes for `PluginContexts`

```
{Locked, Protected, ReadProtected}
```

PluginsLoadingPalette

`PluginsLoadingPalette`

PluginsLoadingPalette[] opens the Plugins Loading palette.

Attributes for `PluginsLoadingPalette`

```
{Locked, Protected, ReadProtected}
```

ProgrammingPalette

ProgrammingPalette

ProgrammingPalette[] opens the programming palette.

Attributes for `ProgrammingPalette`

```
{Locked, Protected, ReadProtected}
```

Purge$Path

Purge$Path

Purge$Path[] returns the value of $Path to what it was just after the Diary.m package was loaded.

Attributes for `Purge$Path`

```
{Locked, Protected, ReadProtected}
```

RecentDiaries

RecentDiaries

RecentDiaries[] gives a list of the recently opened diaries. The form of the list is {{Date, FileName}..} where Date is the
most recently opened and FileName is the full path to the Diary notebook.

Attributes for `RecentDiaries`

```
{Locked, Protected, ReadProtected}
```

RecentNotebooks

RecentNotebooks

RecentNotebooks[] gives a list of the recently opened notebooks. The form of the list is {{Date, FileName}..} where Da
notebook was most recently opened and FileName is the full path to the notebook.

Attributes for `RecentNotebooks`

```
{Locked, Protected, ReadProtected}
```

RecordLength

`RecordLength`

RecordLength[name] is the number of fields in a record of the database given by name.

Attributes for `RecordLength`

```
{Locked, Protected, ReadProtected}
```

RefreshDashboard

`RefreshDashboard`

RefreshDashboard is an option for Dashboard that determines whether any existing Dashboard notebooks are refreshed
executed. Note that, depending on the elements of the Dashboard, some might be refreshed independent of the value f

Attributes for `RefreshDashboard`

```
{Locked, Protected, ReadProtected}
```

RefreshOpenPalettes

`RefreshOpenPalettes`

RefreshOpenPalettes[] refreshes all open palettes associated with this package.

Attributes for `RefreshOpenPalettes`

```
{Locked, Protected, ReadProtected}
```

ReloadDatabase

`ReloadDatabase`

ReloadDatabase[name] reloads the database name if it has already been loaded.

Attributes for `ReloadDatabase`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

RemoveAllWhiteSpace

`RemoveAllWhiteSpace`

RemoveAllWhiteSpace[str] removes all WhiteSpace characters from the string str. WhiteSpace characters are those liste
$WhiteSpaceCharacters.

Attributes for `RemoveAllWhiteSpace`

```
{Locked, Protected, ReadProtected}
```

RemoveBracketingWhiteSpace

`RemoveBracketingWhiteSpace`

RemoveBracketingWhiteSpace[str] removes all WhiteSpace characters from the left and right side of the string str. Whi
those listed in $WhiteSpaceCharacters.

Attributes for `RemoveBracketingWhiteSpace`

```
{Locked, Protected, ReadProtected}
```

RemovePackage

`RemovePackage`

RemovePackage["contextString"] removes the definitions associated with the package contextString. RemovePackage d
that are loaded by the package contextString. Also if the package has any functions or parameters that have the Locke
parameters are not removed.

Attributes for `RemovePackage`

```
{Locked, Protected, ReadProtected}
```

RemoveRedundantDatabaseRecords

`RemoveRedundantDatabaseRecords`

RemoveRedundantDatabaseRecords[name] removes redundant records in the database. For a large database this may tal generally a database is not permitted to have redundant records this function should not generally need to be used.

Default options for `RemoveRedundantDatabaseRecords`

```
{SameTest → Automatic}
```

Attributes for `RemoveRedundantDatabaseRecords`

```
{HoldFirst, Protected, ReadProtected}
```

RemoveSpuriousPalettes

`RemoveSpuriousPalettes`

RemoveSpuriousPalettes[] removes any spurious palettes that may have been created. This is to address a bug in the Ma Mathematica. This is caused because the operating system hides palettes whenever the Mathematica application is not RemoveSpuriousPalettes[] can be executed by the user whenever she or he wishes. It is automatically executed by the places so that spurious palettes are removed if they have arisen. RemoveSpuriousPalettes returns the number of spuric removed.

Attributes for `RemoveSpuriousPalettes`

```
{Locked, Protected, ReadProtected}
```

RemoveTheCellTags

`RemoveTheCellTags`

RemoveTheCellTags is an option to DivideCellAtLineBreaks which specifies whether to remove all of the cell tags of th default value is True. A value of False will yield cells that all have the same cell tags as the original.

Attributes for `RemoveTheCellTags`

```
{Locked, Protected, ReadProtected}
```

ResetCurrentDiary

`ResetCurrentDiary`

ResetCurrentDiary[] resets each of the values of $CurrentDiaryNotebook,$CurrentDiaryNotebookFile,and $CurrentDiar
None.

Attributes for `ResetCurrentDiary`

```
{Locked, Protected, ReadProtected}
```

ResetDefaultParameterValue

`ResetDefaultParameterValue`

ResetDefaultParameterValue[param] resets the value of the parameter to its default if it has a value and if it is in the Dia

Attributes for `ResetDefaultParameterValue`

```
{HoldAll, Locked, Protected, ReadProtected}
```

ResetDiaryNotebookDefaults

`ResetDiaryNotebookDefaults`

ResetDiaryNotebookDefaults[] resets externally adjustable parameters that pertain to Diary function to their default valu
opened its parameters are reset to their defaults. The list of the names of these parameters is given by executing
DiaryNotebookParametersAndFunctions[].

Attributes for `ResetDiaryNotebookDefaults`

```
{Locked, Protected, ReadProtected}
```

RestoreDatabase

`RestoreDatabase`

RestoreDatabase[name] restores the database to the form that it was the last time LoadDatabase or ReloadDatabase was
records will be restored and any new records will be abandoned. Modified records will be replaced with their original

Attributes for `RestoreDatabase`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

RevealTagDate

`RevealTagDate`

RevealTagDate[tag,True] reveals the time associated with the tag "tag" in the cell's CellLabel for the currently selected c
notebook. RevealTagDate[tag,False] removes this display. The tag "tag" should be a string. RevealTagDate[nb,tag,Tr
RevealTagDate[nb,tag,False] have the described function for the notebook object nb.

Attributes for `RevealTagDate`

```
{Locked, Protected, ReadProtected}
```

RightIndentCell

`RightIndentCell`

RightIndentCell[nb] indents the text of the selected cell in the notebook nb from the right by the amount $IndentCellDef
does this for the current InputNotebook[].

Attributes for `RightIndentCell`

```
{Locked, Protected, ReadProtected}
```

RightOutdentCell

`RightOutdentCell`

RightOutdentCell[] outdents the text of the selected cell in the notebook nb from the right by the amount $IndentCellDe
does this for the current InputNotebook[].

Attributes for `RightOutdentCell`

```
{Locked, Protected, ReadProtected}
```

RSSFeedsPalette

RSSFeedsPalette

RSSFeedsPalette[] opens the RSS Palette.

Attributes for `RSSFeedsPalette`

```
{Locked, Protected, ReadProtected}
```

SaveBackupToolbarCell

SaveBackupToolbarCell

SaveBackupToolbarCell[] gives the cell expression for a toolbar cell with "Save", "Backup", and "ShowBackups" buttoı

Attributes for `SaveBackupToolbarCell`

```
{Locked, Protected, ReadProtected}
```

SavedButtonInformation

SavedButtonInformation

SavedButtonInformation["string"] gives the a list of button parameters suitable for use in the custom palettes.

Attributes for `SavedButtonInformation`

```
{}
```

SaveDiary

SaveDiary

SaveDiary[] saves the current Diary. By default all Cells are locked that are of a style contained in the list $LockingCell
locks all cells and closes all open subgroups associated with dates other than the current date. The DefaultCodeCell's
CellOpen->False.

Attributes for SaveDiary

```
{Locked, Protected, ReadProtected}
```

SaveSlideShow

SaveSlideShow

SaveSlideShow is an option to CreateSlideShowFromNotebook and CreateSlideShowFromDiary that determines wheth
should be automatically be saved to the DiaryNotebooksDirectory[].

Attributes for SaveSlideShow

```
{Locked, Protected, ReadProtected}
```

ScaleImage

ScaleImage

ScaleImage is an option to PlaceImage that determines how to scale the width of the resulting image. Its default value is
width equal to $DefaultImageWidth. If its value is a positive integer then the width will be that value. If its value is a
fraction then the image's original width will be scaled by that factor. To make the image display in its original size ch

Attributes for ScaleImage

```
{Locked, Protected, ReadProtected}
```

ScanAllNotebookCells

ScanAllNotebookCells

ScanAllNotebookCells[nb,function] scans the cells of nb and executes "function" at each cell.

Attributes for ScanAllNotebookCells

```
{HoldRest, Locked, Protected, ReadProtected}
```

SearchEngine

SearchEngine

SearchEngine is an option to WebSearch that specifies the name of the search engine to use. The search engines that We
listed in $WebSearchEngines. If an unknown search engine is specified then Google is used by default.

Attributes for `SearchEngine`

```
{Locked, Protected, ReadProtected}
```

SelectionsCellTags

`SelectionsCellTags`

SelectionsCellTags[nb] gives the CellTags that the current selection has in the NotebookObject nb.

Default options for `SelectionsCellTags`

```
{ExcludedTags → {*[*], After*, CellGUID*, DefaultCodeCell, DiaryDate*, Done*,
   DropReissPiece, DueDate*, EmailBodyCell*, EmailBottomCell*, EmailCell*, EmailNo
   EmailToolbarCell*, EntryToolbarCell, EphemeralGUID*, EssayBodyCell*, EssayBotto
   EssayCell*, EssayNotesCell*, EssayToolbarCell*, GUIDTag*, NotebookBackups,
   SaveBackupToolbarCell*, SubjectCell*, ToCell*, ToDo*, TopToolbarCell, CreatedOn
   ConvertedOn, Info*, ToDoPriority*, Archived*, ArchiveDiary, ComputationCellOper
   ComputationCellOperatorName, ComputationCellOperatorTemplate, IntegerQ[ToExpre
 IncludedTags → {*}, NotebooksCellTagsOutput → Automatic, Method → Automatic}
```

Attributes for `SelectionsCellTags`

```
{Protected, ReadProtected}
```

SelectRulesFromList

`SelectRulesFromList`

SelectRulesFromList[list] selects those elements of the list that are rules.

Attributes for `SelectRulesFromList`

```
{Locked, Protected, ReadProtected}
```

SetAttributesString

`SetAttributesString`

SetAttributesString["form", attr] adds attr to the list of attributes of all symbols whose names match any of the string pat

Attributes for `SetAttributesString`

```
{Locked, Protected, ReadProtected}
```

SetAutoSectionHeading

SetAutoSectionHeading

SetAutoSectionHeading[True|False] is used to set the value of $AutoSectionHeading.

Attributes for `SetAutoSectionHeading`

```
{Locked, Protected, ReadProtected}
```

SetBlogTemplateFileName

SetBlogTemplateFileName

SetBlogTemplateFileName[blogName,"name"] sets the value of the blog template for the indicated blog. Available blog
the list returned from BlogTemplates[].

Attributes for `SetBlogTemplateFileName`

```
{Locked, Protected, ReadProtected}
```

SetCellOptions

SetCellOptions

SetCellOptions[nb,opts] sets the options opts for the selected cell in the notebook nb.

Default options for `SetCellOptions`

```
{FilterOptions → True}
```

Attributes for `SetCellOptions`

```
{Protected, ReadProtected}
```

SetCurrentDiary

`SetCurrentDiary`

SetCurrentDiary[nb] sets the current Diary to the notebook object nb. If nb is not a Diary then an error message is gener
returned.

Attributes for `SetCurrentDiary`

```
{Locked, Protected, ReadProtected}
```

SetCurrentDiaryDirectoryAndFile

`SetCurrentDiaryDirectoryAndFile`

SetCurrentDiaryDirectoryAndFile[] does what it says. It is mostly intended as an internal function.

Attributes for `SetCurrentDiaryDirectoryAndFile`

```
{Locked, Protected, ReadProtected}
```

SetDefaultDiaryDirectory

`SetDefaultDiaryDirectory`

SetDefaultDiaryDirectory[directory] sets the value of $DefaultDiaryDirectory to be "directory". SetDefaultDiaryDirecto
$DefaultDiaryDirectory to be the same as the current value of $CurrentDiaryNotebookDirectory.

Attributes for `SetDefaultDiaryDirectory`

```
{Locked, Protected, ReadProtected}
```

SetDefaultPaletteSet

`SetDefaultPaletteSet`

SetDefaultPaletteSet[{paletteName,...}] sets the value of $DefaultPaletteSet. Each of the paletteNames should be the He
The list of Palette Functions is given by

{AdditionalToolsPalette, AllPalettesPalette, AnalyticsPalette, BlogPalette, ComputationPalette, Custom1Palette, Custor

Custom4Palette, Custom5Palette, Custom6Palette, DatabasesPalette, DiaryAccessPalette, DiaryEntriesPalette, DiaryE
DiaryHeadingsPalette, DiaryListPalette, DiaryTemplatesPalette, EmailPalette, EssayPalette, EvaluationPalette, Favor
FormattingPalette, FormTemplatesPalette, NotebooksPalette, NotebookStylesPalette, OrganizationsPalette, OtherFile
PackageProgrammingPalette, PackagesPalette, PluginsLoadingPalette, ProgrammingPalette, RSSFeedsPalette, StyleS
TaggingPalette, ToDosPalette, UnHidePalette, WebSearchPalette, WorkFlowsPalette, WorkLifeToolsPalette, Favorit
AllOpenNotebooksPalette}.

Attributes for `SetDefaultPaletteSet`

```
{Locked, Protected, ReadProtected}
```

SetDefaults

SetDefaults

SetDefaults[] resets externally adjustable parameters to their package default values.Note that generally this function sho
you should set specific parameters in the Diary notebook's DefaultCodeCell.

Attributes for `SetDefaults`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

SetDiaryFilesSearchDepth

SetDiaryFilesSearchDepth

SetDiaryFilesSearchDepth[n] sets the value of $DiaryFilesSearchDepth to be the positive integer n. n should not be chos
number. Its default value is 1. Use SetDiaryFilesSearchDepth to change its value. Generally this value should be 1 or
to difficult-to-track directory and palette listings.

Attributes for `SetDiaryFilesSearchDepth`

```
{Locked, Protected, ReadProtected}
```

SetDiaryNotebookDefault

SetDiaryNotebookDefault

SetDiaryNotebookDefault[] sets the current Diary notebook to have the package defaults.SetDiaryNotebookDefault[note
current Diary notebook to have the notebook options given by notebookOptions.

Attributes for `SetDiaryNotebookDefault`

```
{Locked, Protected, ReadProtected}
```

SetDiaryNotebookDirectory

`SetDiaryNotebookDirectory`

SetDiaryNotebookDirectory[] opens a file dialog that allows you to set the current Diary notebook's directory. It also set
directory to that directory.

Attributes for `SetDiaryNotebookDirectory`

```
{Locked, Protected, ReadProtected}
```

SetDiaryNotebookFile

`SetDiaryNotebookFile`

SetDiaryNotebookFile[] opens a dialog box that allows you to set the current value of the Diary notebook file and assigr
parameter $CurrentDiaryNotebookFile. SetDiaryNotebookFile[NotebookFile] sets the Diary notebook to "NotebookF
should be a string giving the full path to the intended notebook or a list of directories leading to the notebook in quest
argument the name of the notebook itself.

Attributes for `SetDiaryNotebookFile`

```
{Locked, Protected, ReadProtected}
```

SetMathematicaUsageDatabaseTruncationParameters

`SetMathematicaUsageDatabaseTruncationParameters`

SetMathematicaUsageDatabaseTruncationParameters[timeLimit,timeincrement] sets the $MathematicaUsageDatabaseT
$MathematicaUsageDatabaseTimeLimitIncrement parameters. These parameters govern when the WorkLife FrameW
MathematicaUsageDatabase is archived and how much of it is archived. If needed, archiving takes place when the Wo
loaded. SetMathematicaUsageDatabaseTruncationParameters[Default] resets these parameters to their default values

Attributes for `SetMathematicaUsageDatabaseTruncationParameters`

```
{Locked, Protected, ReadProtected}
```

SetMathematicaUsageDatabaseTruncationParametersDialog

`SetMathematicaUsageDatabaseTruncationParametersDialog`

SetMathematicaUsageDatabaseTruncationParametersDialog[] opens a dialog that lets you set the $MathematicaUsageD
$MathematicaUsageDatabaseTimeLimitIncrement parameters. These parameters govern when the WorkLife FrameW
MathematicaUsageDatabase is archived and how much of it is archived. If needed, archiving takes place when the Wo
loaded. Use the function SetMathematicaUsageDatabaseTruncationParameters[Default] to reset these parameters to t

Attributes for `SetMathematicaUsageDatabaseTruncationParametersDialog`

```
{Locked, Protected, ReadProtected}
```

ShowComputationData

`ShowComputationData`

ShowComputationData[tag_String] creates a notebook with the data from the current Diary notebook that are in data cel

Attributes for `ShowComputationData`

```
{Locked, Protected, ReadProtected}
```

ShowDefaultCodeCell

`ShowDefaultCodeCell`

ShowDefaultCodeCell[True|False] opens the DefaultCodeCell if the argument is True and closes it if the argument is Fa

Attributes for `ShowDefaultCodeCell`

```
{Locked, Protected, ReadProtected}
```

ShowDones

`ShowDones`

ShowDones[] creates and displays a notebook containing all the former "ToDo" items that have been marked "Done" fro
notebook.

Default options for ShowDones

```
{ShowEmptyNotebook → True}
```

Attributes for ShowDones

```
{Protected, ReadProtected}
```

ShowDonesSorted

ShowDonesSorted

ShowDonesSorted[] creates and displays a notebook containing all the former "ToDo" items that have been marked "Do
notebook sorted by the date the Dones were marked Done.

Default options for ShowDonesSorted

```
{ShowEmptyNotebook → True}
```

Attributes for ShowDonesSorted

```
{Protected, ReadProtected}
```

ShowEmailLabels

ShowEmailLabels

ShowEmailLabels is an option for EmailNetwork that determines whether the nodes n the network graph should be labe
corresponding email addresses.

Attributes for ShowEmailLabels

```
{Locked, Protected, ReadProtected}
```

ShowEmailsFromNotebook

ShowEmailsFromNotebook

ShowEmailsFromNotebook[nb] opens a notebook containing the emails from the notebook nb.

Attributes for `ShowEmailsFromNotebook`

```
{Locked, Protected, ReadProtected}
```

ShowEmptyNotebook

`ShowEmptyNotebook`

ShowEmptyNotebook is an option for ShowToDos, ShowDones, ShowToDosSortedByDate,ShowToDosSortedByPrior
which determines whether the notebook should be displayed if there are no ToDos or Dones of the specified types in t

Attributes for `ShowEmptyNotebook`

```
{Locked, Protected, ReadProtected}
```

ShowTaggedCells

`ShowTaggedCells`

ShowTaggedCells[nb,tag,True|False] either shows or hides the cells that have been tagged with the tag "tag" according
argument is True or False.

Attributes for `ShowTaggedCells`

```
{Locked, Protected, ReadProtected}
```

ShowToDos

`ShowToDos`

ShowToDos[] creates and displays a notebook containing all the current "ToDo" items from the current Diary notebook.
notebook containing only those ToDos of priority n.

Default options for `ShowToDos`

```
{ShowEmptyNotebook → True}
```

Attributes for `ShowToDos`

```
{Protected, ReadProtected}
```

ShowToDosSortedByDate

ShowToDosSortedByDate

ShowToDosSortedByDate[] creates and displays a notebook containing all the current "ToDo" items from the current D
date the ToDos were created.

Default options for ShowToDosSortedByDate

```
{ShowEmptyNotebook → True}
```

Attributes for ShowToDosSortedByDate

```
{Protected, ReadProtected}
```

ShowToDosSortedByPriority

ShowToDosSortedByPriority

ShowToDosSortedByPriority[] creates and displays a notebook containing all the current "ToDo" items from the curren
their priority.

Default options for ShowToDosSortedByPriority

```
{ShowEmptyNotebook → True}
```

Attributes for ShowToDosSortedByPriority

```
{Protected, ReadProtected}
```

ShowToHideCells

ShowToHideCells

ShowToHideCells[nb,True|False] either shows or hides the cells that have been tagged as "ToHide" according to wheth
True or False.

Attributes for ShowToHideCells

```
{Locked, Protected, ReadProtected}
```

SimpleDialog

SimpleDialog

SimpleDialog[buttons:{{String,Function}...},windowTitle,cellText,opts] allows the entry of a single string. When used
MakeGeneralPalette the Function arguments should be presented in the form Function[...] rather than in the & form.
SimpleDialog[buttons:{{String,Function}...},windowTitle,cellText1,cellText2,opts] allows the entry of two strings.

Default options for `SimpleDialog`

```
{CharacterCheck → True, IncludeInputCell → True, NotebookClose → True}
```

Attributes for `SimpleDialog`

```
{Protected, ReadProtected}
```

SkinParametersTool

SkinParametersTool

SkinParametersTool[] is a tool that allows you to experiment with changing text and background colors of the buttons ir
background colors of some other entities.

Attributes for `SkinParametersTool`

```
{Locked, Protected, ReadProtected}
```

StyleSheetsPalette

StyleSheetsPalette

StyleSheetsPalette[] opens up the Style Sheets palette.

Attributes for `StyleSheetsPalette`

```
{Locked, Protected, ReadProtected}
```

SystemAndMathematicaInformation

SystemAndMathematicaInformation

SystemAndMathematicaInformation[] opens a notebook with informative data on the system that you are running Math
your Mathematica distribution, and on the version of the WorkLife FrameWork™ that you have installed.
SystemAndMathematicaInformation[List] give s you this information in the form of a List.

Attributes for `SystemAndMathematicaInformation`

```
{Locked, Protected, ReadProtected}
```

TagCell

`TagCell`

TagCell[nb,tag] adds the CellTag ToString[tag] to the currently selected cell in the notebook nb where tag is not a string
expression. By default the option IncludeDate has the value IncludeDate→True. This causes the actual tag to be of the
TagCell does not check whether the given tag has already been added to the cell, so multiple copies of the tag can be
may not be the behavior that is desired. When tag is a string then the IncludeDate option has no effect and the string i
for the cell as is. A related function is AddCellTag.

Default options for `TagCell`

```
{IncludeDate → True}
```

Attributes for `TagCell`

```
{Protected, ReadProtected}
```

TaggingPalette

`TaggingPalette`

TaggingPalette[{tags}] opens the tagging palette. "{tags}" is a list of tags. Each tag must be a string. TaggingPalette[] is
TaggingPalette[$TaggingList]. Although the TaggingPalette has a refresh button it is not listed in the list $Refreshabl
the user has control over when she or he wants to alter $TaggingList.

Attributes for `TaggingPalette`

```
{Locked, Protected, ReadProtected}
```

TheNotebookDirectory

`TheNotebookDirectory`

TheNotebookDirectory[nb] gives the directory where the notebook nb is located. TheNotebookDirectory[] gives the dir

TheNotebookDirectory[] is evaluated in is located.

Attributes for `TheNotebookDirectory`

```
{Locked, Protected, ReadProtected}
```

TheNotebookFilePath

`TheNotebookFilePath`

TheNotebookFilePath[nb] gives the full file path to the notebook nb.

Attributes for `TheNotebookFilePath`

```
{Locked, Protected, ReadProtected}
```

TimeEstimate

`TimeEstimate`

TimeEstimate is the head of a function indicating how long the specified ToDo is estimated to take. It appears in the Cel
entered through the ToDosEntryDialog or the MarkToDoEntryDialog.

Attributes for `TimeEstimate`

```
{Locked, Protected, ReadProtected}
```

TimeForm

`TimeForm`

TimeForm is an option to functions that display a date that determine whether the time of day is expressed in military ti
clock or in terms of AM/PM. The default value is TimeForm→12. To express in terms of a 24 hour clock use TimeFo
will default to a 12 hour clock.

Attributes for `TimeForm`

```
{Locked, Protected, ReadProtected}
```

TimeTaken

`TimeTaken`

TimeTaken is the head of a function indicating how long the specified ToDo took to complete. It appears in the CellTag
entered through the MarkDoneEntryDialog.

Attributes for `TimeTaken`

```
{Locked, Protected, ReadProtected}
```

ToDo

`ToDo`

ToDo is a function head indicating a date tag for a ToDo.It is also used as a cell tag for items that have been marked as a

Attributes for `ToDo`

```
{Locked, Protected, ReadProtected}
```

ToDoPriority

`ToDoPriority`

ToDoPriority is used as a cell tag.

Attributes for `ToDoPriority`

```
{Locked, Protected, ReadProtected}
```

ToDosPalette

`ToDosPalette`

ToDosPalette[] opens the ToDos Palette.

Attributes for ToDosPalette

```
{Locked, Protected, ReadProtected}
```

ToDoStatistics

ToDoStatistics

ToDoStatistics[] generates a list of the number of ToDos of each priority in the current Diary notebook. ToDoStatistics[
notebook with the data along with a bar graph of the same.

Attributes for ToDoStatistics

```
{Locked, Protected, ReadProtected}
```

ToggleCellBracket

ToggleCellBracket

ToggleCellBracket[nb] toggles the cell bracket of the selected cell visible or invisible.

Attributes for ToggleCellBracket

```
{Locked, Protected, ReadProtected}
```

ToggleCellFrame

ToggleCellFrame

ToggleCellFrame[nb,linewidth] toggles the cell frame of the selected cell on and off. The cell frame's line width is giver
be a number or True or False.

Attributes for ToggleCellFrame

```
{Locked, Protected, ReadProtected}
```

ToggleCellPrivacy

ToggleCellPrivacy

ToggleCellPrivacy[nb] takes the currently selected Cell in the notebook nb and changes it's font to something that is not

Attributes for `ToggleCellPrivacy`

```
{Locked, Protected, ReadProtected}
```

ToggleDashboard

`ToggleDashboard`

ToggleDashboard[] toggles the dashboard between visible and invisible.

Attributes for `ToggleDashboard`

```
{Locked, Protected, ReadProtected}
```

ToggleDefaultNewCellStyle

`ToggleDefaultNewCellStyle`

ToggleDefaultNewCellStyle[] toggles the default style of a new cell in the current Diary notebook between "Text" and "
default is different than either of these then the default is set to "Text." ToggleDefaultNewCellStyle[nb] toggles the d
the notebook given by the notebook object nb.

Attributes for `ToggleDefaultNewCellStyle`

```
{Locked, Protected, ReadProtected}
```

ToggleEssayNotesCellBodyCell

`ToggleEssayNotesCellBodyCell`

ToggleEssayNotesCellBodyCell[nb] toggles an EssayNotesCell into an EssayBodyCell and vice versa.

Attributes for `ToggleEssayNotesCellBodyCell`

```
{Locked, Protected, ReadProtected}
```

ToggleFunction

`ToggleFunction`

ToggleFunction is an option to ToggleVariable. The value of ToggleFunction should be Automatic, the head of a functic
pure function of one variable. When a value other than the default assignment ToggleVariable→Automatic is assignec
apply the indicated function to the new choice prior to assigning the result to the variable. See the usage message for

Attributes for `ToggleFunction`

`{Locked, Protected, ReadProtected}`

ToggleInputCellsOpenClosed

`ToggleInputCellsOpenClosed`

ToggleInputCellsOpenClosed[True|False] opens or closes the input cells in the current Diary notebook.

Attributes for `ToggleInputCellsOpenClosed`

`{Locked, Protected, ReadProtected}`

ToggleNotebookWindowElements

`ToggleNotebookWindowElements`

ToggleNotebookWindowElements[nb,element] toggles the WindowElements option for the notebook nb with regard to
"element." "element" can be one of "HorizontalScrollBar", "MagnificationPopUp", "StatusArea" and "VerticalScrolI
ToggleNotebookWindowElements[nb] toggles the window element "VerticalScrollBar."

Attributes for `ToggleNotebookWindowElements`

`{Locked, Protected, ReadProtected}`

ToggleSaveBackupToolbarCellOpenClosed

`ToggleSaveBackupToolbarCellOpenClosed`

ToggleSaveBackupToolbarCellOpenClosed[nb] toggles between an opened or closed SaveBackupToolbarCell in the no
one. If the notebook does not have a SaveBackupToolbarCell then ToggleSaveBackupToolbarCellOpenClosed adds a

SaveBackupToolbarCell to the top of the notebook. ToggleSaveBackupToolbarCellOpenClosed has no effect on Diar
SaveBackupToolbarCell to the notebook execute AddSaveBackupToolbarCell[nb].

Attributes for `ToggleSaveBackupToolbarCellOpenClosed`

```
{Locked, Protected, ReadProtected}
```

ToggleShowCellTags

`ToggleShowCellTags`

ToggleShowCellTags[nb] toggles the ShowCellTags option for the notebook object nb between True and False thus resp
hiding the CellTags. ToggleShowCellTags[] toggles the CellTags for InputNotebook[].

Attributes for `ToggleShowCellTags`

```
{Locked, Protected, ReadProtected}
```

ToggleVariable

`ToggleVariable`

ToggleVariable[var,{choices...}] cyclically toggles amongst the elements of the list of choices assigning var to each one
ToggleFunction option to ToggleVariable gives a function to apply to the new choice prior to assigning the result to tl

Default options for `ToggleVariable`

```
{ToggleFunction → Automatic}
```

Attributes for `ToggleVariable`

```
{HoldAll, Protected, ReadProtected}
```

Toggle$LoadPlugins

`Toggle$LoadPlugins`

Toggle$LoadPlugins[] changes the value of $LoadPlugins to True if it is False and to False if it is True and caches the re

Attributes for `Toggle$LoadPlugins`

```
{Locked, Protected, ReadProtected}
```

UnloadDatabase

UnloadDatabase

UnloadDatabase[name] clears (unloads) the database "name" if it has been loaded. Database records are not affected.

Attributes for `UnloadDatabase`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

UnloadMathematicaUsageDatabase

UnloadMathematicaUsageDatabase

UnloadMathematicaUsageDatabase[] unloads (clears) the MathematicaUsageDatabase from memory.

Attributes for `UnloadMathematicaUsageDatabase`

```
{Locked, Protected, ReadProtected}
```

UnlockCell

UnlockCell

UnlockCell[] unlocks the specific cells that are chosen in the current Diary notebook. UnlockCell[nb] unlocks the specif
the notebook nb.

Attributes for `UnlockCell`

```
{Locked, Protected, ReadProtected}
```

UnlockCells

UnlockCells

UnlockCells[] unlocks all of the cells of the types listed in $LockingCellStyles in the current Diary notebook. UnlockCe
the current Diary notebook.

Default options for `UnlockCells`

```
{OpenDefaultCodeCell → False}
```

Attributes for `UnlockCells`

```
{Protected, ReadProtected}
```

UpdateNotebookOrganizationFlow

`UpdateNotebookOrganizationFlow`

UpdateNotebookOrganizationFlow[nb,{"flowtag1","flowtag2",...}] updates the OrganizationFlow of the notebook nb. It
OrganizationFlow to be the list of tags given.

Attributes for `UpdateNotebookOrganizationFlow`

```
{Locked, Protected, ReadProtected}
```

UpdateOrganizationRules

`UpdateOrganizationRules`

UpdateOrganizationRules[organization,{"ruleName1":→rule1,...}] updates the rules for the organization according to the
Method option can be chosen as

Replace: replaces the rules

AppendTo: appends the new rules to the current set

PrependTo: prepends the new rules to the current set

The value Automatic is equivalent to the value Replace. The rules must be int he form of delayed rules with strings as tl

Default options for `UpdateOrganizationRules`

```
{Method → Replace}
```

Attributes for `UpdateOrganizationRules`

```
{Protected, ReadProtected}
```

Update$Databases

Update$Databases

Update$Databases[] updates $Databases so that any databases that no longer exist are removed. Update$Databases[True
DatabasesPalette after $Databases is updated.

Attributes for Update$Databases

```
{Locked, Protected, ReadProtected}
```

Update$FavoriteDiaries

Update$FavoriteDiaries

Update$FavoriteDiaries[] updates all of the information associated with $FavoriteDiaries. Update$FavoriteDiaries also
FavoritesAndRecentPalette if it is open. Update$FavoriteDiaries[False] does not update the FavoritesAndRecentPalet

Attributes for Update$FavoriteDiaries

```
{Locked, Protected, ReadProtected}
```

Update$FavoriteNotebooks

Update$FavoriteNotebooks

Update$FavoriteNotebooks[] updates all of the information associated with $FavoriteNotebooks. Update$FavoriteNoteb
FavoritesAndRecentPalette if it is open. Update$FavoriteNotebooks[False] does not update the FavoritesAndRecentP

Attributes for Update$FavoriteNotebooks

```
{Locked, Protected, ReadProtected}
```

Update$FileSets

Update$FileSets

Update$FileSets[] updates and stores $FileSets for future use. This function us called internally from other $FileSets ma
generally does not need to be called by the user.

Attributes for `Update$FileSets`

```
{Locked, Protected, ReadProtected}
```

Update$RecentDiaries

`Update$RecentDiaries`

Update$RecentDiaries[] updates all of the information associated with $RecentDiaries. Update$RecentDiaries also refre
FavoritesAndRecentPalette if it is open. Update$RecentDiaries[False] does not update the FavoritesAndRecentPalette

Attributes for `Update$RecentDiaries`

```
{Locked, Protected, ReadProtected}
```

Update$RecentNotebooks

`Update$RecentNotebooks`

Update$RecentNotebooks[] updates all of the information associated with $RecentNotebooks. Update$RecentNotebook
FavoritesAndRecentPalette if it is open. Update$RecentNotebooks[False] does not update the FavoritesAndRecentPa

Attributes for `Update$RecentNotebooks`

```
{Locked, Protected, ReadProtected}
```

Update$TaggingList

`Update$TaggingList`

Update$TaggingList[tf] updates the value of $TaggingList and saves it to the TaggingList.m file in $DiaryPackageSetti
argument, tf, determines whether or not the TaggingPalette should be refreshed. Its value should be either True or Fal
Update$TaggingList[{tags...},tf], replaces the tags with those in the list of its first argument. Update$TaggingList[] is
Update$TaggingList[True]. Update$TaggingList[{tags}] is equivalent to Update$TaggingList[{tags},True].

Attributes for `Update$TaggingList`

```
{Locked, Protected, ReadProtected}
```

UsageCellList

UsageCellList

UsageCellList[function, {cellStyle1,cellStyle2,cellStyle3}] gives a list of cells with the usage message for the function. function in question or the name of the function as a string. cellStyle1 is the style of the cell with the functions name a of the cell with the function's usage message.

UsageCellList[function] uses the default values {"Subsection","ObjectName","Usage"} as the cell styles. The "ObjectN styles are present in Mathematica's standard HelpBrowser style sheet, but is \!\(* StyleBox["not",

FontSlant->"Italic"]\) present in a number of other style sheets.

Attributes for `UsageCellList`

```
{HoldFirst, Locked, Protected, ReadProtected}
```

WebSearch

WebSearch

WebSearch["text"] sends the specified text to a web search engine. The name of the search engine that is used is given b The search engines that WebSearch knows about are listed in $WebSearchEngines. WebSearch[nb] searches for the c in the notebook nb.

Default options for `WebSearch`

```
{SearchEngine → Google, FullString → True}
```

Attributes for `WebSearch`

```
{Protected, ReadProtected}
```

WebSearchPalette

WebSearchPalette

WebSearchPalette[] opens the Web Search Palette.

Attributes for `WebSearchPalette`

```
{Locked, Protected, ReadProtected}
```

WhiteSpaceQ

WhiteSpaceQ

WhiteSpaceQ[str] returns True if the string str contains any WhiteSpace characters. WhiteSpace characters are those lis
$WhiteSpaceCharacters.

Attributes for `WhiteSpaceQ`

```
{Locked, Protected, ReadProtected}
```

WorkFlow

WorkFlow

WorkFlow[name,{"flownameA","flownameB",...}] executes the WorkFlow with the given name with the flow list in th
WorkFlow[name] gives the list of delayed rules corresponding to the WorkFlow. WorkFlow[name,Default] executes
order (the order of the rules given in the WorkFlow's original definition: i.e., in the order of the rules in WorkFlow[na
WorkFlow[name,Names] gives the list of the Names of the WorkFlow elements.

Attributes for `WorkFlow`

```
{Locked, Protected, ReadProtected}
```

WorkFlows

WorkFlows

WorkFlows[] gives a list of the available WorkFlows

Attributes for `WorkFlows`

```
{Locked, Protected, ReadProtected}
```

WorkFlowsPalette

WorkFlowsPalette

WorkFlowsPalette[] opens the WorkFlows Palette.

Attributes for `WorkFlowsPalette`

```
{Locked, Protected, ReadProtected}
```

WorkLife

`WorkLife`

WorkLife -- Information on A WorkLife FrameWork can be found on http://scientificarts.com/worklife. This version of
only works for Mathematica versions 5.1 and 5.2 and above. A version compatible with Mathematica Versions 6 and
from http://scientificarts.com/worklife/.

Attributes for `WorkLife`

```
{Locked, Protected, ReadProtected}
```

WorkLifeFrameWorkLicenseAgreement

`WorkLifeFrameWorkLicenseAgreement`

WorkLifeFrameWorkLicenseAgreement[] displays the License Agreement for the use of the Worklife FrameWork™ Sc
and any supporting files.

Attributes for `WorkLifeFrameWorkLicenseAgreement`

```
{Locked, Protected, ReadProtected}
```

WorkLifeSkins

`WorkLifeSkins`

WorkLifeSkins[] gives a list of the available names for WorkLife skins.

Attributes for `WorkLifeSkins`

```
{Locked, Protected, ReadProtected}
```

WorkLifeToolsPalette

`WorkLifeToolsPalette`

WorkLifeToolsPalette[] creates a palette with a default set of WorkLife tools. If your WorkLife Tools Palette is either h
bring it forward by executing WorkLifeToolsPalette[]. Additional buttons can be appended to this palette by assigning
$WorkLifeToolsPaletteExtraButtons and executing WorkLifeToolsPalette[Sequence@@$WorkLifeToolsPaletteExtr

Attributes for `WorkLifeToolsPalette`

```
{Locked, Protected, ReadProtected}
```

$AdditionalToolsPaletteExtraButtons

`$AdditionalToolsPaletteExtraButtons`

$AdditionalToolsPaletteExtraButtons is a list of button parameters that define additional buttons to append to the Additi
format of the list is {{_String,_Function,{___?OptionQ}}...}.

Attributes for `$AdditionalToolsPaletteExtraButtons`

```
{}
```

$AdditionalToolsPaletteWindowMargins

`$AdditionalToolsPaletteWindowMargins`

$AdditionalToolsPaletteWindowMargins specifies the default window margins of the Additional Tools Palette.

Attributes for `$AdditionalToolsPaletteWindowMargins`

```
{}
```

$AddSaveBackupToolbarCellToNewNotebook

`$AddSaveBackupToolbarCellToNewNotebook`

$AddSaveBackupToolbarCellToNewNotebook determines whether an SaveBackupToolbarCell is prepended to a new n
created using NewNotebook. Its default value is True.

Attributes for `$AddSaveBackupToolbarCellToNewNotebook`

```
{}
```

$AlertDialogWindowMargins

`$AlertDialogWindowMargins`

$AlertDialogWindowMargins specifies the default window margins of an Alert box.

Attributes for `$AlertDialogWindowMargins`

`{}`

$AllOpenNotebooksPaletteWindowMargins

`$AllOpenNotebooksPaletteWindowMargins`

$AllOpenNotebooksPaletteWindowMargins specifies the default window margins of the AllOpenNotebooksPalette.

Attributes for `$AllOpenNotebooksPaletteWindowMargins`

`{}`

$AllPalettesPaletteExtraButtons

`$AllPalettesPaletteExtraButtons`

$AllPalettesPaletteExtraButtons is a list of button parameters that define additional buttons to append to the AllPalettes
list is {{_String,_Function|None,{___?OptionQ}}...}.

Attributes for `$AllPalettesPaletteExtraButtons`

`{}`

$ArchiveDiaryDialogNotebook

`$ArchiveDiaryDialogNotebook`

$ArchiveDiaryDialogNotebook is the notebook object for the currently opened Archive Diary Dialog.

Attributes for `$ArchiveDiaryDialogNotebook`

```
{}
```

$ArchiveDiaryDialogWindowMargins

`$ArchiveDiaryDialogWindowMargins`

$ArchiveDiaryDialogWindowMargins specifies the default window margins of the Archive Diary Dialog.

Attributes for `$ArchiveDiaryDialogWindowMargins`

```
{}
```

$AuthorToolsPaletteFile

`$AuthorToolsPaletteFile`

$AuthorToolsPaletteFile gives the Mathematica Author Tools palette file location.

Attributes for `$AuthorToolsPaletteFile`

```
{}
```

$AutoSectionHeading

`$AutoSectionHeading`

$AutoSectionHeading is a parameter that determines whether a section heading is automatically generated in a Diary if
generated on that date. Its default value is True. Its value should be changed with SetAutoSectionHeading.

Attributes for `$AutoSectionHeading`

```
{}
```

$BlogPaletteWindowMargins

`$BlogPaletteWindowMargins`

$BlogPaletteWindowMargins specifies the default window margins of the BlogPalette.

Attributes for `$BlogPaletteWindowMargins`

```
{}
```

$BlogTemplatesDirectory

`$BlogTemplatesDirectory`

$BlogTemplatesDirectory is the directory where blog templates are stored.

Attributes for `$BlogTemplatesDirectory`

```
{Locked, Protected, ReadProtected}
```

$BlueColor

`$BlueColor`

$BlueColor is the color used for the "Blue" button on the Formatting palette.

Attributes for `$BlueColor`

```
{}
```

$CloseAllPalettesPaletteWhenPaletteClicked

`$CloseAllPalettesPaletteWhenPaletteClicked`

$CloseAllPalettesPaletteWhenPaletteClicked is a parameter that determines whether the AllPalettesPalette closes autom
opened from one of its buttons. The default value of $CloseAllPalettesPaletteWhenPaletteClicked is False.

Attributes for `$CloseAllPalettesPaletteWhenPaletteClicked`

```
{}
```

$CloseFavoritePalettesPaletteWhenPaletteClicked

`$CloseFavoritePalettesPaletteWhenPaletteClicked`

$CloseFavoritePalettesPaletteWhenPaletteClicked is a parameter that determines whether the FavoritePalettesPalette clo
palette is opened from one of its buttons. The default value of $CloseFavoritePalettesPaletteWhenPaletteClicked is F

Attributes for `$CloseFavoritePalettesPaletteWhenPaletteClicked`

```
{}
```

$ComputationDataEntryGridDialogWindowMargins

`$ComputationDataEntryGridDialogWindowMargins`

$ComputationDataEntryGridDialogWindowMargins specifies the default window margins of the ComputationDataEntr

Attributes for `$ComputationDataEntryGridDialogWindowMargins`

```
{}
```

$ComputationDataNotebook

`$ComputationDataNotebook`

$ComputationDataNotebook is the current notebook containing a list of data generated from ShowComputationData[tag

Attributes for `$ComputationDataNotebook`

```
{}
```

$ComputationPaletteWindowMargins

`$ComputationPaletteWindowMargins`

$ComputationPaletteWindowMargins specifies the default window margins of the ComputationPalette.

Attributes for `$ComputationPaletteWindowMargins`

```
{}
```

$ContractDiaryNotebookOnSaveDiary

`$ContractDiaryNotebookOnSaveDiary`

$ContractDiaryNotebookOnSaveDiary determines whether Diary cells are contracted when SaveDiary or CloseDiary is is True and it is reset to its default value whenever a Diary is closed. Thus it can be set to False on a Diary by Diary b "$ContractDiaryNotebookOnSaveDiary=False;" in a DefaultCodeCell of the given Diary.

Attributes for `$ContractDiaryNotebookOnSaveDiary`

```
{}
```

$CurrentDiaryNotebook

`$CurrentDiaryNotebook`

$CurrentDiaryNotebook is the notebook object for the currently opened Diary notebook. Open the Diary Notebook usin

Attributes for `$CurrentDiaryNotebook`

```
{}
```

$CurrentDiaryNotebookDirectory

`$CurrentDiaryNotebookDirectory`

$CurrentDiaryNotebookDirectory is the current Diary notebook's directory.

Attributes for `$CurrentDiaryNotebookDirectory`

```
{}
```

$CurrentDiaryNotebookFile

`$CurrentDiaryNotebookFile`

$CurrentDiaryNotebookFile is the value of the current Diary notebook file. You should use SetDiaryNotebook to chang

Attributes for `$CurrentDiaryNotebookFile`

```
{}
```

$CurrentDirectoriesAndFilesNotebook

`$CurrentDirectoriesAndFilesNotebook`

$CurrentDirectoriesAndFilesNotebook is the notebook object for the currently opened current directories and files wind

Attributes for `$CurrentDirectoriesAndFilesNotebook`

```
{}
```

$CurrentPackageNotebook

`$CurrentPackageNotebook`

$CurrentPackageNotebook is the notebook object for the currently opened package notebook.

Attributes for `$CurrentPackageNotebook`

```
{}
```

$CurrentPackageNotebookFile

`$CurrentPackageNotebookFile`

$CurrentPackageNotebookFile is the notebook file path for the most recently opened package notebook.

Attributes for `$CurrentPackageNotebookFile`

```
{}
```

$CustomWorkLifeFrameWork

`$CustomWorkLifeFrameWork`

$CustomWorkLifeFrameWork is a parameter that tells whether the version of the WorkLife FrameWork™ that you are

Attributes for `$CustomWorkLifeFrameWork`

```
{Locked, Protected, ReadProtected}
```

$Databases

`$Databases`

$Databases is a list of databases known to this package. It is in the form of a list with each element a list of length two. '
entry is the name of the database and the second entry is the full path name to the database. $Databases should genera
user.

Attributes for `$Databases`

```
{}
```

$DatabasesPaletteWindowMargins

`$DatabasesPaletteWindowMargins`

$DatabasesPaletteWindowMargins specifies the default window margins of the Database Palette.

Attributes for `$DatabasesPaletteWindowMargins`

```
{}
```

$DateOrder

`$DateOrder`

$DateOrder globally determines the value of the DateOrder option. See the usage message for DateOrder for possible va

Attributes for `$DateOrder`

```
{}
```

$DateTagDiaryOnSaveDiary

`$DateTagDiaryOnSaveDiary`

$DateTagDiaryOnSaveDiary determines whether Diary cells are date tagged when SaveDiary or CloseDiary is executed and it is reset to its default value whenever a Diary is closed. Thus it can be set to False on a Diary by Diary basis by i "$DateTagDiaryOnSaveDiary=False;" in a DefaultCodeCell of the given Diary.

Attributes for `$DateTagDiaryOnSaveDiary`

```
{}
```

$DefaultBlogTemplateFileName

`$DefaultBlogTemplateFileName`

$DefaultBlogTemplateFileName is the default value of the blog template file name. Change its value for a given blog us SetBlogTemplateFileName[blogName,"name"].

Attributes for `$DefaultBlogTemplateFileName`

```
{}
```

$DefaultDiaryDirectory

`$DefaultDiaryDirectory`

$DefaultDiaryDirectory is the default diary directory. You can set the value for this directory by using the SetDefaultDi: you have never set the value of $DefaultDiaryDirectory or if you execute ClearDefaultDiaryDirectory then its value v

Attributes for `$DefaultDiaryDirectory`

```
{}
```

$DefaultImageWidth

`$DefaultImageWidth`

$DefaultImageWidth is the width of an image placed by PlaceImage if the ScaleImage option has its default value of Au

Attributes for `$DefaultImageWidth`

```
{}
```

$DeleteBlogEntryCreateNotebook

`$DeleteBlogEntryCreateNotebook`

$DeleteBlogEntryCreateNotebook determines whether or not a notebook containing the deleted Blog Entry will be creat
False.

Attributes for `$DeleteBlogEntryCreateNotebook`

```
{}
```

$DeleteDefaultCodeCellBeforeApplyDiaryTemplateToDiary

`$DeleteDefaultCodeCellBeforeApplyDiaryTemplateToDiary`

$DeleteDefaultCodeCellBeforeApplyDiaryTemplateToDiary determines whether the current DefaultCodeCells in the D
applying a template through use of ApplyDiaryTemplateToDiary["name"].

Attributes for `$DeleteDefaultCodeCellBeforeApplyDiaryTemplateToDiary`

```
{}
```

$DeleteEmailCreateNotebook

`$DeleteEmailCreateNotebook`

$DeleteEmailCreateNotebook determines whether or not a notebook containing the deleted email will be created. Its de

Attributes for `$DeleteEmailCreateNotebook`

```
{}
```

$DeleteEssayCreateNotebook

`$DeleteEssayCreateNotebook`

$DeleteEssayCreateNotebook determines whether or not a notebook containing the deleted Essay will be created. Its de

Attributes for `$DeleteEssayCreateNotebook`

```
{}
```

$DiaryAccessPaletteWindowMargins

`$DiaryAccessPaletteWindowMargins`

$DiaryAccessPaletteWindowMargins specifies the default window margins of the DiaryAccessPalette.

Attributes for `$DiaryAccessPaletteWindowMargins`

```
{}
```

$DiaryEntriesPaletteWindowMargins

`$DiaryEntriesPaletteWindowMargins`

$DiaryEntriesPaletteWindowMargins specifies the default window margins of the DiaryEntriesPalette.

Attributes for `$DiaryEntriesPaletteWindowMargins`

```
{}
```

$DiaryEntryPaletteWindowMargins

`$DiaryEntryPaletteWindowMargins`

$DiaryEntryPaletteWindowMargins specifies the default window margins of the DiaryEntryPalette.

Attributes for `$DiaryEntryPaletteWindowMargins`

```
{}
```

$DiaryFilesSearchDepth

`$DiaryFilesSearchDepth`

$DiaryFilesSearchDepth is an integer that specifies how many directories deep DiaryFiles looks to find named Diary no
1. Use SetDiaryFilesSearchDepth to change its value. Generally this value should be 1 or 2, Other values may lead to
and palette listings.

Attributes for `$DiaryFilesSearchDepth`

```
{}
```

$DiaryFunctionsNotebook

`$DiaryFunctionsNotebook`

$DiaryFunctionsNotebook is the current notebook containing a list of functions from this package.

Attributes for `$DiaryFunctionsNotebook`

```
{}
```

$DiaryHeadingsDialogWindowMargins

`$DiaryHeadingsDialogWindowMargins`

$DiaryHeadingsDialogWindowMargins specifies the default window margins of the Diary Headings Dialog.

Attributes for `$DiaryHeadingsDialogWindowMargins`

```
{}
```

$DiaryHeadingsPaletteWindowMargins

`$DiaryHeadingsPaletteWindowMargins`

$DiaryHeadingsPaletteWindowMargins specifies the default window margins of the Diary Headings Palette.

Attributes for `$DiaryHeadingsPaletteWindowMargins`

```
{}
```

$DiaryHeadingsSection

`$DiaryHeadingsSection`

$DiaryHeadingsSection is the list of current diary Section headings.

Attributes for `$DiaryHeadingsSection`

```
{}
```

$DiaryHeadingsSubsection

`$DiaryHeadingsSubsection`

$DiaryHeadingsSubsection is the list of current diary Subsection headings.

Attributes for `$DiaryHeadingsSubsection`

```
{}
```

$DiaryHeadingsSubsubsection

`$DiaryHeadingsSubsubsection`

$DiaryHeadingsSubsubsection is the list of current Diary Subsubsection headings.

Attributes for `$DiaryHeadingsSubsubsection`

```
{}
```

$DiaryKeywords

`$DiaryKeywords`

$DiaryKeywords is the internal parameter that contains the diary keywords.

Attributes for `$DiaryKeywords`

```
{}
```

$DiaryListPaletteAutoClose

`$DiaryListPaletteAutoClose`

$DiaryListPaletteAutoClose is a parameter that determines whether the Diary List palette automatically closes when a D
value is True.

Attributes for `$DiaryListPaletteAutoClose`

```
{}
```

$DiaryListPaletteWindowMargins

`$DiaryListPaletteWindowMargins`

$DiaryListPaletteWindowMargins specifies the default window margins of the DiaryListPalette.

Attributes for `$DiaryListPaletteWindowMargins`

```
{}
```

$DiaryNotebookDefaultOptions

`$DiaryNotebookDefaultOptions`

$DiaryNotebookDefaultOptions is a list of the default options for Diary notebooks.

Attributes for `$DiaryNotebookDefaultOptions`

```
{}
```

$DiaryNotebookStyle

`$DiaryNotebookStyle`

$DiaryNotebookStyle gives the style for a new Diary. It must be a string chosen from the list $DiaryNotebookStyles.

Attributes for `$DiaryNotebookStyle`

```
{}
```

$DiaryNotebookStyles

`$DiaryNotebookStyles`

$DiaryNotebookStyles is a list of possible style definitions for new Diaries.

Attributes for `$DiaryNotebookStyles`

```
{}
```

$DiaryOtherNotebookDefaultOptions

`$DiaryOtherNotebookDefaultOptions`

$DiaryOtherNotebookDefaultOptions is a list of the default options for some notebooks associated with this package.

Attributes for `$DiaryOtherNotebookDefaultOptions`

```
{}
```

$DiaryPackageLoaded

`$DiaryPackageLoaded`

$DiaryPackageLoaded has the value True if the Diary package has been loaded.

Attributes for `$DiaryPackageLoaded`

```
{Locked, Protected, ReadProtected}
```

$DiaryPackageRootDirectory

`$DiaryPackageRootDirectory`

$DiaryPackageRootDirectory should be the directory where the Diary.m package is located. If it is not located there ther
not been properly installed. Execute FileNameNoDirectory/@FileNames["*" ,$DiaryPackageRootDirectory] to deter

Diary.m is present.

Attributes for `$DiaryPackageRootDirectory`

```
{Locked, Protected, ReadProtected}
```

$DiaryPackageSettingsDirectory

`$DiaryPackageSettingsDirectory`

$DiaryPackageSettingsDirectory should be the directory where the files with settings for the state of the Diary.m packag

Attributes for `$DiaryPackageSettingsDirectory`

```
{Locked, Protected, ReadProtected}
```

$DiaryPluginsDirectory

`$DiaryPluginsDirectory`

$DiaryPluginsDirectory should be the directory where the plug-ins for the Diary.m package are located.

Attributes for `$DiaryPluginsDirectory`

```
{Locked, Protected, ReadProtected}
```

$DiaryResourcesBackupDirectory

`$DiaryResourcesBackupDirectory`

$DiaryResourcesBackupDirectory should be the directory where backups of the resources for the Diary.m package are l

Attributes for `$DiaryResourcesBackupDirectory`

```
{Locked, Protected, ReadProtected}
```

$DiaryResourcesDirectory

`$DiaryResourcesDirectory`

$DiaryResourcesDirectory should be the directory where the resources for the Diary.m package are located.

Attributes for `$DiaryResourcesDirectory`

```
{Locked, Protected, ReadProtected}
```

$DiarySkinsDirectory

`$DiarySkinsDirectory`

$DiarySkinsDirectory should be the directory where Diary Skins are located

Attributes for `$DiarySkinsDirectory`

```
{Locked, Protected, ReadProtected}
```

$DiarySubDirectories

`$DiarySubDirectories`

$DiarySubDirectories is a list of the names of potential standard subdirectories of a Diary directory.

Attributes for `$DiarySubDirectories`

```
{Locked, Protected, ReadProtected}
```

$DiaryTemplatesDirectory

`$DiaryTemplatesDirectory`

$DiaryTemplatesDirectory should be the directory where the file templates for the Diary.m package are located.

Attributes for `$DiaryTemplatesDirectory`

```
{Locked, Protected, ReadProtected}
```

$DiaryTemplatesPaletteWindowMargins

`$DiaryTemplatesPaletteWindowMargins`

$DiaryTemplatesPaletteWindowMargins specifies the default window margins of the Diary Templates Palette.

Attributes for `$DiaryTemplatesPaletteWindowMargins`

```
{}
```

$DirectoryBrowser

`$DirectoryBrowser`

$DirectoryBrowser is a parameter that, if True specifies that DirectoryBrowser is used to select a default Diary directory
standard file system dialog is used.

Attributes for `$DirectoryBrowser`

```
{}
```

$DirectoryBrowserExcludedCharacters

`$DirectoryBrowserExcludedCharacters`

$DirectoryBrowserExcludedCharacters is a list of those characters that, if a directory begins with one of those character
will not display that directory in its list of choices. This does not apply to the directory favorites.

Attributes for `$DirectoryBrowserExcludedCharacters`

```
{}
```

$DirectoryBrowserNotebook

`$DirectoryBrowserNotebook`

$DirectoryBrowserNotebook is the notebook Object corresponding to the currently open Directory Browser.

Attributes for `$DirectoryBrowserNotebook`

```
{}
```

$DirectoryBrowserWindowMargins

`$DirectoryBrowserWindowMargins`

$DirectoryBrowserWindowMargins specifies the default window margins of the DirectoryBrowser.

Attributes for `$DirectoryBrowserWindowMargins`

```
{}
```

$DonesNotebook

`$DonesNotebook`

$DonesNotebook is the current notebook containing a list of Dones generated from ShowDones[].

Attributes for `$DonesNotebook`

```
{}
```

$EmailCellBackground

`$EmailCellBackground`

$EmailCellBackground gives the color directive for newly created email cells. Its default value is GrayLevel[.96].

Attributes for `$EmailCellBackground`

```
{}
```

$EmailCellFrameColor

`$EmailCellFrameColor`

$EmailCellFrameColor gives the color directive for the frame of the To, Cc, Bcc, Subject, and Body cells in an email fo

Attributes for `$EmailCellFrameColor`

```
{}
```

$EmailFormCellFrameColor

$EmailFormCellFrameColor

$EmailFormCellFrameColor gives the color directive for the frame of the EmailToolbarCell and EmailBottomCell in an

Attributes for $EmailFormCellFrameColor

```
{}
```

$EmailInCurrentDiary

$EmailInCurrentDiary

$EmailInCurrentDiary determines whether the Email Palette only works with Diaries. If True then the notebook that em
default, $CurrentDiaryNotebook. If False then the email is created and edited in the current InputNotebook[]. The de
$EmailInCurrentDiary=True.

Attributes for $EmailInCurrentDiary

```
{}
```

$EmailNotesCellFrameColor

$EmailNotesCellFrameColor

$EmailNotesCellFrameColor gives the color directive for the frame of a notes cell in an email form.

Attributes for $EmailNotesCellFrameColor

```
{}
```

$EmailPageWidth

$EmailPageWidth

$EmailPageWidth gives the page width of email cells. It's value (the same choices as those of the PageWidth parameter)
PaperWidth, WindowWidth, or Infinity. It's default value is WindowWidth.

Attributes for `$EmailPageWidth`

```
{}
```

$EmailPaletteExtraButtons

`$EmailPaletteExtraButtons`

$EmailPaletteExtraButtons is a list of button parameters that define additional buttons to append to the Email palette. Th
{{_String,_Function,{___?OptionQ}}...}.

Attributes for `$EmailPaletteExtraButtons`

```
{}
```

$EmailPaletteWindowMargins

`$EmailPaletteWindowMargins`

$EmailPaletteWindowMargins specifies the default window margins of the Email Palette.

Attributes for `$EmailPaletteWindowMargins`

```
{}
```

$EmailSignatureString

`$EmailSignatureString`

$EmailSignatureString is a string that contains signature text that is appended to the end of an email. Its default value is

Attributes for `$EmailSignatureString`

```
{}
```

$EnquoteCharacterString

`$EnquoteCharacterString`

$EnquoteCharacterString is the character string used by EnquoteEmailBodyCell. It's default value is ">".

Attributes for `$EnquoteCharacterString`

```
{}
```

$EssayInCurrentDiary

`$EssayInCurrentDiary`

$EssayInCurrentDiary determines whether the Essay Palette only works with Diaries. If True then the notebook that ess
default, $CurrentDiaryNotebook. If False then the essay is created and edited in the current InputNotebook[]. The de
$EssayInCurrentDiary=True.

Attributes for `$EssayInCurrentDiary`

```
{}
```

$EssayPaletteWindowMargins

`$EssayPaletteWindowMargins`

$EssayPaletteWindowMargins specifies the default window margins of the Essay Palette.

Attributes for `$EssayPaletteWindowMargins`

```
{}
```

$EvaluationPaletteWindowMargins

`$EvaluationPaletteWindowMargins`

$EvaluationPaletteWindowMargins specifies the default window margins of the EvaluationPalette notebook.

Attributes for `$EvaluationPaletteWindowMargins`

```
{}
```

$EvaluationTrackingContexts

$EvaluationTrackingContexts

$EvaluationTrackingContexts is a list of contexts that will be used when $SaveEvaluatedToFile is True. These are the c
Saved to $CurrentEvaluationTrackingSaveFile. The default value is {"Global`"}.

Attributes for `$EvaluationTrackingContexts`

```
{}
```

$FavoriteDiaries

$FavoriteDiaries

$FavoriteDiaries

Attributes for `$FavoriteDiaries`

```
{}
```

$FavoriteDiariesOpen

$FavoriteDiariesOpen

$FavoriteDiariesOpen determines whether the Favorite Diaries sub-palette in the Favorites & Recent Palette is open wh
refreshed. Its default value is True.

Attributes for `$FavoriteDiariesOpen`

```
{}
```

$FavoriteNotebooks

$FavoriteNotebooks

$FavoriteNotebooks

Attributes for `$FavoriteNotebooks`

```
{}
```

$FavoriteNotebooksOpen

`$FavoriteNotebooksOpen`

$FavoriteNotebooksOpen determines whether the Favorite Notebooks sub-palette in the Favorites & Recent Palette is o
opened or refreshed. Its default value is True.

Attributes for `$FavoriteNotebooksOpen`

```
{}
```

$FavoritePalettesPaletteWindowMargins

`$FavoritePalettesPaletteWindowMargins`

$FavoritePalettesPaletteWindowMargins specifies the default window margins of the Favorite Palettes Palette.

Attributes for `$FavoritePalettesPaletteWindowMargins`

```
{}
```

$FavoritesAndRecentOrganizerWindowMargins

`$FavoritesAndRecentOrganizerWindowMargins`

$FavoritesAndRecentOrganizerWindowMargins specifies the default window margins of the $FavoritesAndRecentOrga

Attributes for `$FavoritesAndRecentOrganizerWindowMargins`

```
{}
```

$FavoritesAndRecentPaletteAutoClose

`$FavoritesAndRecentPaletteAutoClose`

$FavoritesAndRecentPaletteAutoClose determines whether the Favorites & Recent palette automatically closes when a

Its default value is False.

Attributes for `$FavoritesAndRecentPaletteAutoClose`

```
{}
```

$FavoritesAndRecentPaletteNotebook

`$FavoritesAndRecentPaletteNotebook`

$FavoritesAndRecentPaletteNotebook is the notebook object for the FavoritesAndRecentPalette notebook. Open the Fa
using FavoritesAndRecentPalette[].

Attributes for `$FavoritesAndRecentPaletteNotebook`

```
{}
```

$FavoritesAndRecentPaletteWindowMargins

`$FavoritesAndRecentPaletteWindowMargins`

$FavoritesAndRecentPaletteWindowMargins specifies the default window margins of the FavoritesAndRecentPalette.

Attributes for `$FavoritesAndRecentPaletteWindowMargins`

```
{}
```

$FileSets

`$FileSets`

$FileSets is a list of your Web Sets. Each Web Set in the list is of the form {name, {urls...}}.

Attributes for `$FileSets`

```
{}
```

$FormattingBackgroundColors

`$FormattingBackgroundColors`

$FormattingBackgroundColors is the list of extra colors (along with their names) that appear on the formatting palette. U
AddFormattingBackgroundColors and DeleteFormattingBackgroundColors to change the entries in this list.

Attributes for `$FormattingBackgroundColors`

```
{}
```

$FormattingPaletteExtraButtons

`$FormattingPaletteExtraButtons`

$FormattingPaletteExtraButtons is a list of button parameters that define additional buttons to append to the Formatting
list is {{_String,_Function,{___?OptionQ}}...}.

Attributes for `$FormattingPaletteExtraButtons`

```
{}
```

$FormattingPaletteWindowMargins

`$FormattingPaletteWindowMargins`

$FormattingPaletteWindowMargins specifies the default window margins of the Formatting Palette.

Attributes for `$FormattingPaletteWindowMargins`

```
{}
```

$FormattingTextColors

`$FormattingTextColors`

$FormattingTextColors is the list of extra colors (along with their names) that appear on the formatting palette. Use Add
DeleteFormattingTextColors to change the entries in this list.

Attributes for `$FormattingTextColors`

```
{}
```

$FunctionsWithOptions

$FunctionsWithOptions

$FunctionsWithOptions gives a list of the names of functions in this package for which Options[funct]=!={}.

Attributes for `$FunctionsWithOptions`

```
{Locked, Protected, ReadProtected}
```

$FunctionsWithUsageMessages

$FunctionsWithUsageMessages

$FunctionsWithUsageMessages gives a list of functions and parameters in this package that have usage messages. Gene
do not have usage messages should be considered internal to the package and are not intended to be used by the user u
documented to be safe for such use.

Attributes for `$FunctionsWithUsageMessages`

```
{Locked, Protected, ReadProtected}
```

$GreenColor

$GreenColor

$GreenColor is the color used for the "Green" button on the Formatting palette.

Attributes for `$GreenColor`

```
{}
```

$HideAllPalettesExcludedPalettes

$HideAllPalettesExcludedPalettes

$HideAllPalettesExcludedPalettes is a list of those Palettes that should not be hidden when executing HideAllPalettes[].
Strings that are the window titles of the palettes that you do not want hidden. A list of all such window titles can be fo
$PaletteWindowTitles. The default value is $HideAllPalettesExcludedPalettes={}.

Attributes for `$HideAllPalettesExcludedPalettes`

```
{}
```

$HTMLSaveTemplatesDirectory

`$HTMLSaveTemplatesDirectory`

$HTMLSaveTemplatesDirectory is the directory where HTMLSaveWithTemplate templates are stored.

Attributes for `$HTMLSaveTemplatesDirectory`

```
{Locked, Protected, ReadProtected}
```

$IncrementalSave

`$IncrementalSave`

$IncrementalSave determines whether Evaluation Tracking will save parameters as they are added with no further revisi
initially assigned values but not any later changes) or with a complete revision at each evaluation (thus capturing thei
default value is $IncrementalSave→True.

Attributes for `$IncrementalSave`

```
{}
```

$IndentCellDefault

`$IndentCellDefault`

$IndentCellDefault is the parameter by which IndentCell and OutdentCell (and RightIndentCell and RightOutdentCell)
indentation.

Attributes for `$IndentCellDefault`

```
{}
```

$KeywordEntryDialogWindowMargins

`$KeywordEntryDialogWindowMargins`

$KeywordEntryDialogWindowMargins specifies the default window margins of the KeywordEntryDialog.

Attributes for `$KeywordEntryDialogWindowMargins`

```
{}
```

$LoadedDatabases

`$LoadedDatabases`

$LoadedDatabases gives a list of the names of those databases that have been loaded in the current session.

Attributes for `$LoadedDatabases`

```
{}
```

$LoadPlugins

`$LoadPlugins`

$LoadPlugins determines whether plugins are loaded automatically by the package when it is first loaded. Its default val
value by executing the function Toggle$LoadPlugins[].

Attributes for `$LoadPlugins`

```
{}
```

$LockCellsOnSaveDiary

`$LockCellsOnSaveDiary`

$LockCellsOnSaveDiary determines whether Diary cells are locked when SaveDiary or CloseDiary is executed. It's defa
reset to its default value whenever a Diary is closed. Thus it can be set to False on a Diary by Diary basis by including
"$LockCellsOnSaveDiary=False;" in a DefaultCodeCell of the given Diary.

Attributes for `$LockCellsOnSaveDiary`

```
{}
```

$LockingCellStyles

`$LockingCellStyles`

$LockingCellStyles is a list of those cell styles that are locked when SaveDiary[], CloseDiary[], or LockCells[] is execu

Attributes for `$LockingCellStyles`

```
{}
```

$MacintoshAppTruncation

`$MacintoshAppTruncation`

$MacintoshAppTruncation is a parameter that determines (when the operating system is $System==="Mac OS X") or $
x86 (32-bit)" whether OpenFileOrDirectory adjusts its argument so that, if a path to the contents of an application (a M
Package) is given, then the path to the top level ".app" will be substituted so that the application will be opened when
executed.

Attributes for `$MacintoshAppTruncation`

```
{}
```

$MakeTargetBlankForHTMLSaveWithTemplate

`$MakeTargetBlankForHTMLSaveWithTemplate`

$MakeTargetBlankForHTMLSaveWithTemplate determins whether hyperlinks within the html generated by HTMLSav
new window when clicked upon. Its default value is True.

Attributes for `$MakeTargetBlankForHTMLSaveWithTemplate`

```
{}
```

$MarkToDoEntryDialogWindowMargins

`$MarkToDoEntryDialogWindowMargins`

$MarkToDoEntryDialogWindowMargins specifies the default window margins of the Mark ToDo Entry Dialog.

Attributes for `$MarkToDoEntryDialogWindowMargins`

```
{}
```

$MaximumNumberOfBlogEntries

`$MaximumNumberOfBlogEntries`

$MaximumNumberOfBlogEntries gives the maximum number of blog entries to appear on a blog's front page.

Attributes for `$MaximumNumberOfBlogEntries`

```
{}
```

$MaximumNumberOfRecentPostsLinks

`$MaximumNumberOfRecentPostsLinks`

$MaximumNumberOfRecentPostsLinks gives the maximum number of links to recent posts to appear on a blog's front p
any older ones appear on the Blog's archive page.

Attributes for `$MaximumNumberOfRecentPostsLinks`

```
{}
```

$NewDiaryNotebookDialogWindowMargins

`$NewDiaryNotebookDialogWindowMargins`

$NewDiaryNotebookDialogWindowMargins specifies the default window margins of the New Diary Entry Dialog.

Attributes for `$NewDiaryNotebookDialogWindowMargins`

```
{}
```

$NewNotebookDefaultOptions

`$NewNotebookDefaultOptions`

$NewNotebookDefaultOptions is a list of the default options for new notebooks. In contrast, the options for new Diary r

$DiaryNotebookDefaultOptions.

Attributes for `$NewNotebookDefaultOptions`

```
{}
```

$NewScratchNotebookDefaultOptions

`$NewScratchNotebookDefaultOptions`

$NewScratchNotebookDefaultOptions is a list of the default options for new scratch notebooks. In contrast, the options are governed by $DiaryNotebookDefaultOptions.

Attributes for `$NewScratchNotebookDefaultOptions`

```
{}
```

$NotebookSearchDepth

`$NotebookSearchDepth`

$NotebookSearchDepth is the number of directories below the specified one that NotebookDiscovery[dir] looks to find

Attributes for `$NotebookSearchDepth`

```
{}
```

$NotebooksPaletteWindowMargins

`$NotebooksPaletteWindowMargins`

$NotebooksPaletteWindowMargins specifies the default window margins of the Notebooks Palette.

Attributes for `$NotebooksPaletteWindowMargins`

```
{}
```

$NotebookStylesPaletteWindowMargins

`$NotebookStylesPaletteWindowMargins`

$NotebookStylesPaletteWindowMargins specifies the default window margins of the Notebook Styles Palette.

Attributes for `$NotebookStylesPaletteWindowMargins`

```
{}
```

$NotebookTypes

`$NotebookTypes`

$NotebookTypes gives a list of the possible notebook types that can be returned by NotebookType. This is a list of strin

Attributes for `$NotebookTypes`

```
{Locked, Protected, ReadProtected}
```

$NotesCellFrameColor

`$NotesCellFrameColor`

$NotesCellFrameColor gives the color directive for the frame of a notes cell in Diary or other notebook. CellFrameColo
Essays are governed by $EmailCellFrameColor and $EssayCellFrameColor respectively.

Attributes for `$NotesCellFrameColor`

```
{}
```

$NumberOfFavoriteDiaries

`$NumberOfFavoriteDiaries`

$NumberOfFavoriteDiaries

Attributes for `$NumberOfFavoriteDiaries`

```
{}
```

$NumberOfFavoriteNotebooks

`$NumberOfFavoriteNotebooks`

$NumberOfFavoriteNotebooks

Attributes for `$NumberOfFavoriteNotebooks`

```
{}
```

$NumberOfRecentDiaries

`$NumberOfRecentDiaries`

$NumberOfRecentDiaries

Attributes for `$NumberOfRecentDiaries`

```
{}
```

$NumberOfRecentNotebooks

`$NumberOfRecentNotebooks`

$NumberOfRecentNotebooks

Attributes for `$NumberOfRecentNotebooks`

```
{}
```

$OpenBlogInBrowserUponPublishing

`$OpenBlogInBrowserUponPublishing`

$OpenBlogInBrowserUponPublishing determines whether the blog index page is opened in the default browser after it h
PublishBlogEntry. The default value is True.

Attributes for `$OpenBlogInBrowserUponPublishing`

```
{}
```

$OpenTheDatabasesPalette

`$OpenTheDatabasesPalette`

$OpenTheDatabasesPalette determines whether the Databases Palette is opened when a Diary is chosen from the Diary
value is False.

Attributes for `$OpenTheDatabasesPalette`

```
{}
```

$OpenTheFavoritesAndRecentPalette

`$OpenTheFavoritesAndRecentPalette`

$OpenTheFavoritesAndRecentPalette determines whether the FavoritesAndRecentPalette is opened when a Diary is cho
Palette. Its default value is False.

Attributes for `$OpenTheFavoritesAndRecentPalette`

```
{}
```

$OpenTheNotebooksPalette

`$OpenTheNotebooksPalette`

$OpenTheNotebooksPalette determines whether the Notebooks Palette is opened when a Diary is chosen from the Diary
value is False.

Attributes for `$OpenTheNotebooksPalette`

```
{}
```

$OpenThePackagesPalette

`$OpenThePackagesPalette`

$OpenThePackagesPalette determines whether the Packages Palette is opened when a Diary is chosen from the Diary Li
is False.

Attributes for `$OpenThePackagesPalette`

```
{}
```

$PackageDatabasesDirectory

`$PackageDatabasesDirectory`

$PackageDatabasesDirectory is the directory where package databases are located.

Attributes for `$PackageDatabasesDirectory`

```
{Locked, Protected, ReadProtected}
```

$PackagesPaletteWindowMargins

`$PackagesPaletteWindowMargins`

$PackagesPaletteWindowMargins specifies the default window margins of the Packages Palette.

Attributes for `$PackagesPaletteWindowMargins`

```
{}
```

$PaletteBackgroundColor

`$PaletteBackgroundColor`

$PaletteBackgroundColor is the color used for the backgrounds of the palettes. If you want to have an interesting experi
$PaletteBackgroundColor:=Hue[Random[]]. To return it to its default value, execute ResetDefaultParameterValue[$P

Attributes for `$PaletteBackgroundColor`

```
{}
```

$PaletteNames

`$PaletteNames`

$PaletteNames is a list of the names of the palettes in this package.

Attributes for `$PaletteNames`

```
{Locked, Protected, ReadProtected}
```

$PaletteNotebookNames

`$PaletteNotebookNames`

$PaletteNotebookNames is a list of the names of the palette notebooks in this package.

Attributes for `$PaletteNotebookNames`

```
{Locked, Protected, ReadProtected}
```

$PaletteWindowTitles

`$PaletteWindowTitles`

$PaletteWindowTitles gives a list of the WindowTitles of palettes in this package along with the corresponding palette f

Attributes for `$PaletteWindowTitles`

```
{Locked, Protected, ReadProtected}
```

$PasteDateWindowMargins

`$PasteDateWindowMargins`

$PasteDateWindowMargins specifies the default window margins of the PasteDate notebook.

Attributes for `$PasteDateWindowMargins`

```
{}
```

$PermaLinkCellBackgroundColor

`$PermaLinkCellBackgroundColor`

$PermaLinkCellBackgroundColor gives the Background color for a Blog's permalink cell.

Attributes for `$PermaLinkCellBackgroundColor`

```
{}
```

$PrivacyFontFamily

`$PrivacyFontFamily`

$PrivacyFontFamily is the Font used for ToggleCellPrivacy. Its default value is "Wingdings".

Attributes for `$PrivacyFontFamily`

```
{}
```

$PrivacyFontSize

`$PrivacyFontSize`

$PrivacyFontSize is the FontSize used for ToggleCellPrivacy. Its default value is 4.

Attributes for `$PrivacyFontSize`

```
{}
```

$ProgrammingPaletteWindowMargins

`$ProgrammingPaletteWindowMargins`

$ProgrammingPaletteWindowMargins specifies the default window margins of the Programming Palette.

Attributes for `$ProgrammingPaletteWindowMargins`

```
{}
```

$RecentDiaries

`$RecentDiaries`

$RecentDiaries

Attributes for `$RecentDiaries`

```
{}
```

$RecentDiariesOpen

`$RecentDiariesOpen`

$RecentDiariesOpen determines whether the Recent Diaries sub-palette in the Favorites & Recent Palette is open when
refreshed. Its default value is True.

Attributes for `$RecentDiariesOpen`

```
{}
```

$RecentNotebooks

`$RecentNotebooks`

$RecentNotebooks

Attributes for `$RecentNotebooks`

```
{}
```

$RecentNotebooksOpen

`$RecentNotebooksOpen`

$RecentNotebooksOpen determines whether the Recent Notebooks sub-palette in the Favorites & Recent Palette is oper
opened or refreshed. Its default value is True.

Attributes for `$RecentNotebooksOpen`

```
{}
```

$RedColor

`$RedColor`

$RedColor is the color used for the "Red" button on the Formatting palette.

Attributes for `$RedColor`

```
{}
```

$SaveEvaluatedToFile

`$SaveEvaluatedToFile`

$SaveEvaluatedToFile determines whether definitions of parameters in the contexts give by $EvaluationTrackingConte
$CurrentEvaluationTrackingSaveFile for later reloading if desired. This only works if EvaluationTracking is being us
Palette.

Attributes for `$SaveEvaluatedToFile`

```
{}
```

$SlideShowPaletteFile

`$SlideShowPaletteFile`

$SlideShowPaletteFile gives the Mathematica Slide Show palette file location.

Attributes for `$SlideShowPaletteFile`

```
{}
```

$SortFavoriteDiaries

`$SortFavoriteDiaries`

$SortFavoriteDiaries determines whether favorite diaries are sorted in the FavoritesAndRecentPalette. Its default value i
according to $FavoriteDiariesSortingFunction.

Attributes for `$SortFavoriteDiaries`

```
{}
```

$SortFavoriteNotebooks

`$SortFavoriteNotebooks`

$SortFavoriteNotebooks determines whether favorite notebooks are sorted in the FavoritesAndRecentPalette. Its default
according to $FavoriteNotebooksSortingFunction.

Attributes for `$SortFavoriteNotebooks`

```
{}
```

$StyleSheetsPaletteWindowMargins

`$StyleSheetsPaletteWindowMargins`

$StyleSheetsPaletteWindowMargins specifies the default window margins of the Style Sheets Palette.

Attributes for `$StyleSheetsPaletteWindowMargins`

```
{}
```

$TaggedCellsNotebookDefaultOptions

`$TaggedCellsNotebookDefaultOptions`

$TaggedCellsNotebookDefaultOptions gives a list of options to use for notebooks generated by CreateNotebookFromTa
value is $TaggedCellsNotebookDefaultOptions={}, and it is reset to its default whenever a Diary is loaded or the Tag
(See the usage message for TaggingPalette.) When $TaggedCellsNotebookDefaultOptions has its default value the op
generated by CreateNotebookFromTaggedCells are the same as those of the notebook that the tagged cells originally

Attributes for `$TaggedCellsNotebookDefaultOptions`

```
{}
```

$TaggingList

$TaggingList

$TaggingList

Attributes for `$TaggingList`

```
{}
```

$TaggingPaletteWindowMargins

$TaggingPaletteWindowMargins

$TaggingPaletteWindowMargins specifies the default window margins of the Tagging Palette.

Attributes for `$TaggingPaletteWindowMargins`

```
{}
```

$ToDoColor

$ToDoColor

$ToDoColor gives the color of the text in cells that have been created as or marked as a "ToDo".

Attributes for `$ToDoColor`

```
{}
```

$ToDoDingbat

$ToDoDingbat

$ToDoDingbat is a pure function of one variable that determines the dingbat to use for a ToDo cell. The pure function sl
default value is ("💡"<>ToString[#]&). The parameter in the pure function is the priority number for the cell.

Attributes for `$ToDoDingbat`

```
{}
```

$ToDoEntryDialogWindowMargins

`$ToDoEntryDialogWindowMargins`

$ToDoEntryDialogWindowMargins specifies the default window margins of the ToDos Entry Dialog.

Attributes for `$ToDoEntryDialogWindowMargins`

```
{}
```

$ToDosNotebook

`$ToDosNotebook`

$ToDosNotebook is the current notebook containing a list of ToDos generated from ShowToDos[].

Attributes for `$ToDosNotebook`

```
{}
```

$ToDosPaletteWindowMargins

`$ToDosPaletteWindowMargins`

$ToDosPaletteWindowMargins specifies the default window margins of the ToDos Palette.

Attributes for `$ToDosPaletteWindowMargins`

```
{}
```

$ToolbarCellBackgroundColor

`$ToolbarCellBackgroundColor`

$ToolbarCellBackgroundColor gives the background color of Diary toolbars.

Attributes for `$ToolbarCellBackgroundColor`

```
{}
```

$UnHidePaletteNotebook

`$UnHidePaletteNotebook`

$UnHidePaletteNotebook is the notebook object for the UnHidePalette.

Attributes for `$UnHidePaletteNotebook`

```
{}
```

$WebSearchEngines

`$WebSearchEngines`

$WebSearchEngines gives a list of search engines that WebSearch knows about. Each element in this list is a list of 4 ite

1: The name of the search Engine

2: A pattern of differing choices for the name

3: A query string (URL) expressed as a pure function

4: A default URL expressed as a pure function

To see examples of these you can evaluate $WebSearchEngines. The correct form for the query string varies from searc
and generally can be determined by looking at the URL for a search with the particular search engine in a web browse

Attributes for `$WebSearchEngines`

```
{}
```

$WhiteSpaceCharacters

`$WhiteSpaceCharacters`

$WhiteSpaceCharacters is a list of WhiteSpace characters.

Attributes for `$WhiteSpaceCharacters`

`{Locked, Protected, ReadProtected}`

$WorkLifeFrameWorkPassword

`$WorkLifeFrameWorkPassword`

$WorkLifeFrameWorkPassword gives the Password for this copy of the WorkLife FrameWork™.

Attributes for `$WorkLifeFrameWorkPassword`

`{Locked, Protected, ReadProtected}`

$WorkLifeFrameWorkReleaseNumber

`$WorkLifeFrameWorkReleaseNumber`

$WorkLifeFrameWorkReleaseNumber gives the current release number of this package.

Attributes for `$WorkLifeFrameWorkReleaseNumber`

`{Locked, Protected, ReadProtected}`

$WorkLifeFrameWorkTrialPasswordQ

`$WorkLifeFrameWorkTrialPasswordQ`

$WorkLifeFrameWorkTrialPasswordQ has the value True if the password, $WorkLifeFrameWorkPassword, correspond
version of this software.

Attributes for `$WorkLifeFrameWorkTrialPasswordQ`

`{Locked, Protected, ReadProtected}`

$WorkLifeFrameWorkVersion

`$WorkLifeFrameWorkVersion`

$WorkLifeFrameWorkVersion gives a string containing information on the current version of this package.

Attributes for `$WorkLifeFrameWorkVersion`

```
{Locked, Protected, ReadProtected}
```

$WorkLifeFrameWorkVersionNumber

```
$WorkLifeFrameWorkVersionNumber
```

$WorkLifeFrameWorkVersionNumber gives the current version number of this package.

Attributes for `$WorkLifeFrameWorkVersionNumber`

```
{Locked, Protected, ReadProtected}
```

$WorkLifeToolsPaletteExtraButtons

```
$WorkLifeToolsPaletteExtraButtons
```

$WorkLifeToolsPaletteExtraButtons is a list of button parameters that define additional buttons to append to the WorkL
format of the list is {{_String,_Function|None,{___?OptionQ}}...}.

Attributes for `$WorkLifeToolsPaletteExtraButtons`

```
{}
```

$WorkLifeToolsPaletteWindowMargins

```
$WorkLifeToolsPaletteWindowMargins
```

$WorkLifeToolsPaletteWindowMargins specifies the default window margins of the WorkLifeToolsPalette.

Attributes for `$WorkLifeToolsPaletteWindowMargins`

```
{}
```

["ComputationType"] holds the raw data of data cell of type "ComputationType" after ComputeDiaryNotebook has be
ComputeDiaryNotebook is acting on multiple ComputationTypes then [{"ComputationType1","ComputationType2

Attributes for

```
{}
```