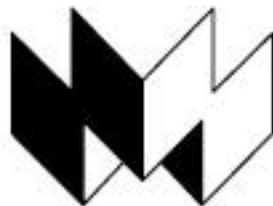


Cumulus 4.0

and

AppleScript



Multi-Ad Services
1720 W. Detweiller Drive
Peoria, IL 61615-1695

Trademark Information

Apple and Macintosh are registered trademarks of Apple Computer, Inc. System 7 and AppleScript are trademarks of Apple Computer, Inc.

APPLE COMPUTER, INC. ("APPLE") MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, REGARDING THE APPLE SOFTWARE. APPLE DOES NOT WARRANT, GUARANTEE OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE APPLE SOFTWARE IN TERM OF ITS CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS OR OTHERWISE. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE APPLE SOFTWARE IS ASSUMED BY YOU. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU.

IN NO EVENT WILL APPLE, ITS DIRECTORS, OFFICERS, EMPLOYEES OR AGENTS BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL OR INDIRECT DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, AND THE LIKE) ARISING OUT OF THE USE OR INABILITY TO USE THE APPLE SOFTWARE EVEN IF APPLE HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE MAY NOT APPLY TO YOU. APPLE'S LIABILITY TO YOU FOR ACTUAL DAMAGES FROM ANY CAUSE WHATSOEVER, AND REGARDLESS OF THE FORM OF THE ACTION (WHETHER IN CONTRAST, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR OTHERWISE), WILL BE LIMITED TO \$50.

Cumulus 4.0.....	i
and.....	i
AppleScript.....	i
Introduction.....	1
Compatibility to Cumulus 3.0.....	2
Changes Made to Object Classes.....	3
Changes Made to Commands.....	3
Compatibility to Cumulus 2.5.....	4
Changes Made to Object Classes.....	4
Changes Made to Commands.....	4
Object Classes.....	5
Object Classes of the Standard Suite.....	5
Object Classes of the Database Suite.....	10
Object Classes of the Cumulus Suite.....	21
Commands.....	29
Commands of the Standard Suite.....	29
Commands of the Database Suite.....	36
Commands of the Cumulus Suite.....	37

Introduction

This documentation is for people who are already familiar with the Apple Scripting System. It describes how to control the Cumulus Desktop PLUS and Cumulus Client applications using AppleScript commands.

For writing a script you need the following:

- Apple Macintosh System 7.1 or greater
- Apple's Scripting System (AppleScript)
- An application that allows writing and executing scripts (e.g. Apple's "Script Editor")

This document describes the AppleScript vocabulary of Cumulus. Cumulus is fully scriptable which means you can write and execute a script to do the things that Cumulus offers through its user interface.

The vocabulary defines objects and commands that extend the AppleScript language already defined by AppleScript and the installed scripting additions.

Some of the objects, properties and commands are defined only for Cumulus Desktop PLUS and can not be used with Cumulus Client.

Cumulus supports most of the "Database AppleEvent Suite" defined by Apple. This suite consists of a description of objects and commands to control a database in general. Such a general database contains one or more tables. Each table contains rows (data records), columns (fields). The intersection of a row and a column is called a cell. It contains the actual value of the record at the corresponding field. All objects can be specified either by index or by name.

The Cumulus database contains two tables: the catalog table containing all visible data records and the categories table containing all categories. As a shortcut you can access the rows, columns and cells of the catalog table directly as elements of the database object.

All AppleScript names are written in *Courier*.

Compatibility to Cumulus 3.0

The version 4.0 of Cumulus adds some new features and properties to existing objects that should not break scripts written for Cumulus 3.0.

Scripts compiled for Cumulus 3.0 should run unchanged.

Cumulus 4.0 now allows remote administration of a server from any client computer. This results in more AppleScript features being available to the Cumulus Client application.

The field "Image Name" is changed to "Record Name". Cumulus 4.0 still accepts the column name "Image Name" but will return "Record Name" whenever it is asked.

Changes Made to Object Classes

application preferences

The property “show record name” of the application preferences was previously called “show image name”.

database settings

A new property “publish to Internet” allows to switch on sharing this catalog to the World Wide Web (see the Cumulus Internet Information Server).

Changes Made to Commands

add database file

This command is new and is part of the remote administration of a Cumulus server.

backup

This command got a new optional parameter to specify the type of backup to create: minimal backup or full backup.

choose server

This new command brings up a dialog to connect to a Cumulus server and returns the server specification.

close database file

This new command is part of the remote administration and closes an open catalog file at the Cumulus server.

new database file

This new command is part of the remote administration and creates a new catalog file at the Cumulus server.

open remote database

This command got a new parameter that specifies the user name for connecting. The preposition for specifying the password has changed.

Compatibility to Cumulus 2.5

The version 3.0 of Cumulus adds some new features and properties to existing objects that should not break scripts written for Cumulus 2.5.

Scripts compiled for Cumulus 2.5 should run unchanged.

Because the term "keyword" is changed to "category" throughout the whole software you will notice that what you knew as "keyword" objects are now called "category" objects when you look at your compiled scripts.

Changes Made to Object Classes

cataloging preferences

The property "keywords" has changed to "categories".

cell

Since cells are named the same as the corresponding column the changes in the column names also apply to cell names.

column

For columns of the catalog table the column name "keywords" has changed to "categories". Please also notice the change in the property names. The old name "keywords" is still accepted by Cumulus 3.0.

For columns of the categories table the name "subkeywords" is changed to "subcategories". Please also notice the change in the property names. The old name "subkeywords" is still accepted by Cumulus 3.0.

row

For rows of the catalog table the property "keywords" has changed to "categories". Please also notice the change in the column names. The old name "keywords" is still accepted by Cumulus 3.0.

For rows of the categories table the property "subkeywords" is changed to "subcategories". Please also notice the change in the column names. The old name "subkeywords" is still accepted by Cumulus 3.0.

table

The two table objects of a Cumulus database are now called "Catalog" and "Categories" but the name "Keywords" is still accepted by Cumulus 3.0. Scripts that access tables by index are not affected.

window

The window kind of the category window has changed from "keywords window" to "category window".

Changes Made to Commands

show categories window

This command was previously called "show keyword window".

Object Classes

Cumulus supports object classes defined by Apple for the standard and the database suite. These object classes are generally used by various applications. These object classes are extended to support additional properties that are specific to Cumulus. This allows you to have direct access to this data instead of having to specify a second object class hierarchy for Cumulus objects.

Cumulus also defines some private object classes in the Cumulus suite.

Object Classes of the Standard Suite

Cumulus supports the following object classes defined by Apple in the Standard suite:

application: The application Cumulus. This object contains all other objects as elements or properties.

selection-object: The currently visible user selection. This object usually contains all the selected objects as its elements.

window: A window of Cumulus. Each window has a property **kind** to identify the type. A window that belongs to a database has a property **database** that references the database object. The record information window has also a property **row** that identifies the row inside the database that is displayed. The find window has a property **search query** which represents the displayed query as text.

application

Description

There is one object describing the application Cumulus itself. All other objects are specified as elements or properties of this object.

Plural Form

applications

Elements

database: An open image database. All open databases are ordered according to the order of their database window in Cumulus from front to back.

window: The windows of Cumulus.

Properties

properties: A record containing all properties of the given object.

best type: The best type representing the application object is **record**. This record contains all properties of the application object.

default type: Same as **best type**.

class: The object class of the **application** object: **application**.

clipboard: The clipboard, currently not implemented.

frontmost: A boolean value indicating whether the application is the frontmost.

name: The name of the application. This is the same name as the name of the application file that was started.

selection: The current selection in the frontmost window, if any. This is the same as the selection property of the frontmost window. Please read the description of the selection in the window object below.

version: The version number of the application as a string.

user interaction: A boolean indicating whether Cumulus should bring up dialogs while processing a command. Error message are always suppressed when Cumulus is controlled by a script but there are other possible dialogs that may appear.

cataloging options: An object of class **cataloging preferences** that contains the current settings made in the Cataloging Options dialog of Cumulus.

preferences: An object of class **application preferences** that contains all setting made in the Preferences dialog in Cumulus. See the **application preferences** object for further information.

selection-object

Description

This object represents the selection that is visible to the user. The actual objects that are selected are accessible as elements of the selection object. You can access the rows, columns and cells of the selection the same way you do with the table object.

Elements

cell: The cells currently selected. This can be cells of the catalog table or the categories table depending of the window the selection has been made.

column: The columns currently selected. This can be columns of the catalog table or the categories table depending of the window the selection has been made.

row: The rows currently selected. This can be rows of the catalog table or the categories table depending of the window the selection has been made.

Properties:

properties: A record containing all properties of the given object.

contents: The contents of the selection. This is a list of row objects for all the objects that are currently selected.

best type: The best type representing the application object is **reference**. When getting the whole selection object Cumulus will return a list of object references to row objects.

class: The object class of the **selection-object: selection-object** .

default type: Same as **best type** .

window

Description

There is one object describing a window of Cumulus. Cumulus extends the definition by some properties that allow direct access to Cumulus specific window information. These properties mostly belong to windows of a certain kind and they are only available for those windows. Whenever you access these properties please make sure that the window is of the necessary kind by checking the property

Plural Form

windows

Properties

properties: A record containing all properties of the given object.

best type: The best type representing the window object is **record**. This record contains all properties of the window object.

bounds: The bounding rectangle of the window in screen coordinates. The bounds are returned as a list of four values in the order left, top, right, bottom.

class: The object class of the **window** object: **window**.

closeable: A boolean indicating whether the window can be closed.

default type: Same as **best type**.

floating: A boolean indicating whether the window is floating in front of all other windows.

index: The number of the window in the order front-to-back starting at 1 for the frontmost window.

modal: A boolean indicating whether the window is modal which means that it has to stay in front until it is closed.

name: The title of the window.

resizable: A boolean indicating whether the window can be resized.

selection: The selection the user made in this window. This is an object of the class selection-object. Cumulus supports rows of the catalog table to be selected in the catalog window and rows of the categories table to be selected in the category window of a database. To select rows please use the select command.

titled: A boolean indicating whether the window has a title bar.

visible: A boolean indicating whether the window is visible. Cumulus allows access only to windows that are visible so this value is always true and can not be changed.

zoomable: A boolean indicating whether the window can be zoomed to fill the whole screen.

zoomed: A boolean indicating whether the window is already zoomed to fill the whole screen.

kind (database window/information window/find window/deliver window/preview window/category window/database setup window/cataloging options window/application preferences window): The kind of window. The following properties are available only for windows of the given kind. Please check this property before accessing the properties below.

view (administration view/text view/thumbnail view). The current view of the catalog window. This property is only available for windows of the `catalog window` kind.

displayed thumbnail size: The current display size for thumbnails in the catalog window. This property is only available for windows of the `catalog window` kind.

native thumbnail size: The thumbnail size as it is set in the database setup dialog at the server. This read-only property allows to get the best viewing size for thumbnails of this particular catalog window. This property is only available for windows of the `catalog window` kind.

keep category names unique: Indicates whether the database should force category names to be unique. This does not change existing category names but will force unique names in the future.

view percentage: The current viewing ratio given as a percentage. This property is only available for windows of the `preview window` kind.

search query: The current query of the find window as a string. This string is of the same format as the export file of a search query. You can also use the value of this property in the find command to perform a search using the search criteria currently visible in the find window of a database. This property is only available for windows of the `find window` kind.

database: The database where the window belongs to (if any). This property is available only for windows of the following kind: `database window/information window/find window/category window/database setup window`.

row: The row of the catalog table that is shown in this information window. This property is only available for windows of the `information window` kind.

Object Classes of the Database Suite

Cumulus supports the following object classes defined by Apple in the Database suite:

cell: A field of a data record or category record. The most important property of it is its value.

column: One of the 30 data fields of a Cumulus database.

database: An image catalog. It contains two tables: catalog and categories. As a shortcut it also contains the rows, columns and cells of the catalog table directly as elements.

row: A data record (catalog table) or category record (categories table) of the database.

table: A table that consists of rows, columns and cells. You do not have to specify rows, columns and cells as elements of the catalog table because they are all also elements of the corresponding database object.

cell

Description

This object is the intersection of a row (record) and a column (field) in a table of a database. The purpose of a cell is to hold the value for the record field. Each cell object contains a fixed set of properties regardless of the table it can be found. Please read the section about the column object above for more information about the available columns in a Cumulus database.

Plural Form

cells

Properties

properties: A record containing all properties of the given object.

best type: The best data type for a cell object: **record**. This record contains all properties of the cell object.

class: The object class for this object: cell.

default type: The same as the default type of the column object the cell belongs to. See the chapter about the columns of the two table for more information.

formula: The formula stored in this cell. Cumulus does not support formulas.

lock: Cumulus does not support locking of cells and therefor always returns no lock.

name: The name of the cell. This is the same as the name of the corresponding column (see above).

protection:

repeat size: The number of values of this cell. Only cells of the column "Categories" of the table "Catalog" can have up to 16383 values. Those cells return a list as their value.

value: The value of the cell. The data type of this value depends on the column the cell belongs to and is the type returned by the **default type** property.

column

Description

This object describes a single field in a table of a database. When getting a whole column object Cumulus returns by default a list of all cells of this column.

Plural Form

columns

Elements

cell: The cells for this column. They are numbered top to bottom as the rows are.

Properties

properties: A record containing all properties of the given object.

access: Cumulus only allows read access to columns.

best type: The best data type for a column object: **record**. This record contains all properties of the column object.

class: The object class for column objects: **column**.

default type: The default type of the data returned when getting the whole column object. It depends on the table and column index. See the chapter about columns of the two tables below.

formula: The formula stored in the column. Cumulus does not store formulas.

lock: Cumulus does not support locking of columns and therefor always returns no lock.

name: The name of the column. This depends on the table that contains the column (see below).

nulls allowed: Indicates whether values of cells of this column can be left empty. Cumulus needs values in the following columns: "Alias", "Thumbnail", "Status", "Cannot Be Modified", "Cannot Be Deleted".

protection: Specifies whether the elements or formula property of the column can be changed. Cumulus allows changing of those values that are modifiable in the record information window of Cumulus.

repeating: Indicates whether it is a repeating column. Only values for the "Categories" column can be repeated.

repeat size: The maximum number of values that can be stored in each cell of this column. Only the cell for the column "Categories" can have up to 16383 values. All other cells only store single values.

unique value: Specifies whether the value of a cell of this column needs to be unique. Cumulus does not force unique values in cells.

Columns of the Catalog Table

The catalog table contains 30 columns representing the 30 fields of the record that Cumulus offers. Some of the columns contain information that can also be found in other columns. For example the "File Size" column contains a value that is the sum of the "File Resource Size" and the "File Data Size". The value of the column "Alias" is also splitted into its parts and those parts are available as values of the following columns: "File Server Zone Name", "File Server Name", "Volume Name", "Folder Names", "File Name". This allows reading and writing of the file name only without having to read and write the full alias.

The following table lists all columns of the catalog table:

	Name	Write Access	Data Type	Description
1	Record Name	•	string	The name of the image
2	File Type	•	string	The file type (4 characters) of the image file
3	Color Mode	•	constant	The color mode of the image (“Line Art”, “Grayscale” or “Color Image”)
4	Horizontal Resolution	•	real	The horizontal resolution of the image
5	Vertical Resolution	•	real	The vertical resolution of the image
6	Record Modification Date	•	date	Modification date of the record
7	File Creator Type	•	string	The signature of the software, that created the image file (4 characters)
8	File Creation Date	•	date	The creation date of the image file
9	File Modification Date	•	date	The modification date of the image file
10	File Data Size	•	integer	The data size of the image file in bytes
11	File Resource Size	•	integer	The resource size of the image file in bytes
12	Width	•	real	The width of the image given in inches
13	Height	•	real	The height of the image given in inches
14	Status	•	string	The status of the image
15	Cannot Be Modified	•	boolean	Record cannot be modified
16	Cannot Be Deleted	•	boolean	Record cannot be deleted
17	User	•	string	The person who cataloged the image
18	Record Creation Date	•	date	The date the record was created (the image was cataloged)
19	Software	•	string	The name of the software that created the image file
20	Thumbnail	•	picture	The thumbnail
21	Alias	•	alias	The reference to the original image file.
22	Notes	•	string	The notes attached to the record
23	Categories	•	list of strings	The list of categories assigned to the record
24	Type Specific	•	string	The type-specific information as a hex string
25	File Server Name	•	string	The name of the file server if the original file was stored on an AppleShare compatible file server. The value of this column is derived from the value of the column "Alias". Setting the value of this column affects the value of the column "Alias".

26	File Server Zone Name	•	string	The name of AppleTalk zone for the file server if the original file was stored on an AppleShare compatible file server. The value of this column is derived from the value of the column "Alias". Setting the value of this column affects the value of the column "Alias".
27	Volume Name	•	string	The name of the volume where the original image file can be found. The value of this column is derived from the value of the column "Alias". Setting the value of this column affects the value of the column "Alias".
28	Folder Names	•	list of strings	The names of the folders (not including the volume name). The value of this column is derived from the value of the column "Alias". Setting the value of this column affects the value of the column "Alias".
29	File Name	•	string	The name of the original file. The value of this column is derived from the value of the column "Alias". Setting the value of this column affects the value of the column "Alias".
30	File Size		integer	The total size of the image file. The value of this column is calculated as the sum of the columns "File Resource Size" and "File Data Size".

Fig. 1 Columns of the Catalog Table

Columns of the Categories Table

The categories table contains all the categories that are defined in the corresponding Cumulus database. The order of the rows is the same as the order in the category window of Cumulus so you can access the category rows by index to follow the order they are displayed in the window. The only difference is that the categories table contains all categories regardless whether they are currently visible or not.

	Name	Write Access	Data Type	Description
1	Name	•	string	The name of the category
2	Subcategories	•	list of integers	A list of row IDs for all sub-categories
3	Parent	•	integer	The row ID of the parent category's row
4	Collapsed	•	boolean	A boolean indicating that the sub-categories are currently not visible in the categories window
5	Level		integer	The level of this category, starting at 1 for the topmost categories

6	Category Type	•	constant	The type of this category: normal category, related category, or folder category. You can turn a category into a normal category by simply modifying this column or property. If you want to change it to a different type of category please modify the columns described below. For related categories you only have to modify the Original Category ID. For folder categories you can either modify both the Volume ID and the Directory ID or modify the Directory which combines them together.
7	Original Category ID	•	integer	For related categories only: the ID of the original category. Modifying this property of a normal category or folder category turns the category into a related category with the given original category id.
8	Volume ID	•	integer	For folder categories only: The ID of the volume that category refers to. If the corresponding volume is mounted and the directory still exists the property Directory contains an alias to the directory.
9	Directory ID	•	integer	For folder categories only: The ID of the directory (folder on a volume) that category refers to. If the corresponding volume is mounted and the directory still exists the property Directory contains an alias to the directory.
10	Directory	•	alias	For folder categories only: The alias of the folder that the category belongs to. This can be derived from the Volume ID and Directory ID above and can only be read if the corresponding volume is mounted. It is most useful for setting the folder that the category belongs to.

Fig. 2 Columns of the Categories Table

database

Description

This object describes an image database. An object of this class contains two tables as well as all rows, columns and cells of the first (catalog) table as elements. When specifying cells by index the cells are ordered row by row.

Elements

table: The table objects of the database. A Cumulus database has two tables: Table 1 is called "Catalog" and contains all records or images of the database. Table 2 is called "Categories" and contains all categories (visible or not visible) of the database. As a shortcut you can omit specifying the catalog table and

row: The currently visible records of the database. These are actually the rows of the table index 1 named "Catalog".

column: The 24 data fields of the Cumulus image database. These are actually the columns of the table index 1 named "Catalog".

cell: The cells for all rows and all columns. These are actually the cells of the table index 1 named "Catalog".

category: Objects of the class **category** representing the top-level categories in the database. Each of the category objects contains its sub-categories as elements.

image status: Objects of the class **image status** representing the status list of the database.

Properties

properties: A record containing all properties of the given object.

best type: The appropriate data type for objects of this class: **database**.

class: The class of this object: **database**.

default type: The default type if the whole object is returned: **record**. This record contains all properties of the database.

lock: The locking state of the database, always "Unlocked".

name: The name of the database.

catalog window: The database window of this database showing the contents of the catalog table.

sorting rows: A boolean indicating that Cumulus is still sorting the rows found. You cannot change the sorting of the catalog table while Cumulus is sorting.

remote database: A boolean indicating whether this database was opened as a remote database through a client-server connection. If the database is a remote database the property **database server** contains the machine specification where the Cumulus server application is running.

database server: The machine where the Cumulus server application is running. Only available for remote databases. Check the value of the property **remote database** prior to accessing this.

database file: This file specification is the database file if the database is not remote. Check the value of the property **remote database** before you access this property.

info: An object of class **database info** containing information about the database file. This is only available for direct databases. Check the value of the property **remote database** before you access this property.

settings: An object of class **database settings** containing the setup of the database file. This is only available for direct databases. Check the value of the property **remote database** before you access this property.

row

Description

This object describes a record of a table of a Cumulus database. A row of the catalog table represents a record entry in the Cumulus catalog whereas a row of the categories table represents a category of the database.

For faster access and readability Cumulus adds properties corresponding to the cells of a row according to the table. The following contains elements and properties that are valid for rows of both tables. Please read the following chapters for additional properties.

Elements

cell: The cells of this row. Please read the chapter about the column object above for the numbering and values of the cells.

Properties

properties: A record containing all properties of the given object.

name: The name of the image or category.

id: A unique number for this record returned as a string.

class: The object class for this object: row.

best type: The best data type for a row object: **record**. This record contains all properties of the row object.

access: According to the values of the cells "Cannot Be Modified" and "Cannot Be Deleted" Cumulus allows updating, changing, deleting or only reading of this record.

lock: Cumulus does not support locking of rows and therefor always returns no lock.

protection: Specifies whether the values of cells of this record can be modified. Cumulus returns "Read Only" or "Read/Write" value according to the value of the cell "Cannot Be Modified".

Additional Properties of Rows of the Catalog Table

For convenience Cumulus supports additional properties for rows of the catalog table that represent the cells of the row. Accessing properties of a row instead of cells makes your script shorter and easier to read. The properties are named exactly as the cells (and columns) they represent. See the column object for a complete list of all columns and their names. There are only the following exceptions: The property of the cell "Record Name" is called **name**. The property of the cell "Alias" is called **file reference**. Please read the chapter about the columns of the catalog table for more information.

file type: The file type (4 characters) of the image file

color mode: The color mode of the image ("Line Art", "Grayscale" or "Color Image")

horizontal resolution: The horizontal resolution of the image in dots per inch. If the image is resolution independent, the horizontal resolution is 0.

vertical resolution: The vertical resolution of the image in dots per inch. If the image is resolution independent, the vertical resolution is 0.

record modification date: Modification date of the record.

creator type: The signature of the software, that created the image file (string of 4 characters).

file creation date: The creation date of the image file.

file modification date: The modification date of the image file.

file data size: The data size of the image file in bytes.

file resource size: The resource size of the image file in bytes.

width: The width of the image given in the units specified in the application preferences. If the units are set to pixels and the image is resolution independent, the width is measured in inch.

height: The height of the image given in the units specified in the application preferences. If the units are set to pixels and the image is resolution independent, the height is measured in inch

status: The status of the image as a string.

cannot be modified: A boolean indicating whether the row is protected and can not be modified.

cannot be deleted: A boolean indicating whether the row is protected and can not be deleted.

user: The name of the person who cataloged the image.

record creation date: The date the record was created (the image was cataloged).

software: The name of the software that created the image file.

thumbnail: The thumbnail as a picture.

file reference: The reference to the original image file.

notes: The notes attached to the record.

categories: The categories assigned to the record as a list of strings.

type specific: The type-specific information as a hex string.

file server name: The name of the file server if the original file was stored on an AppleShare compatible file server. The value of this property is derived from the value of the **file reference** property. Setting the value of this property affects the value of the **file reference** property.

file server zone name: The name of AppleTalk zone for the file server if the original file was stored on an AppleShare compatible file server. The value of this property is derived from the value of the **file reference** property. Setting the value of this property affects the value of the **file reference** property.

volume name: The name of the volume where the original image file can be found. The value of this property is derived from the value of the **file reference** property. Setting the value of this property affects the value of the **file reference** property.

NOTE: If you have installed a Scripting Addition called "Set Volume" you can not use the term "set volume name to..." AppleScript will stop checking the script text with an error message because the term is already defined by the Scripting Addition. Please use the following instead: "set (volume name) to..."

folder names: The names of the folders (not including the volume name). The value of this property is derived from the value of the **file reference** property. Setting the value of this property affects the value of the **file reference** property.

file name: The name of the original file. The value of this column is derived from the value of the **file reference** property. Setting the value of this property affects the value of the **file reference** property.

file size: The total size of the image file. The value of this property is calculated as the sum of the **file data size** and **file resource size** properties.

Additional Properties of Rows in the Categories Table

For convenience Cumulus supports additional properties for rows of the categories table that represent the cells of the row. Accessing properties of a row instead of cells makes your script shorter and easier to read. The properties are named exactly as the cells they represent. Please read the chapter about the columns of the categories table for more information.

name: The name of the category.

subcategories: A list of row IDs of all sub-categories.

parent: The row ID of the parent category's row. A 0 indicates that this category is at top level.

collapsed: A boolean indicating that the sub-categories are currently not visible in the categories window

level: The level of this category, starting at 1 for the topmost categories

category type: The kind of this category: `normal category`, `related category`, or `folder category`. These category types have additional properties for additional data: related categories offer the ID of the original category to be read and written. Folder categories allow you to read and write the folder reference either by ID or by an alias. See details below.

original category ID: For related categories only: the ID of the original category. Modifying this property of a normal category or folder category turns the category into a related category with the given original category id.

volume ID: For folder categories only: This is the ID of the volume that the folder category refers to. Together with the `directory ID` below this is a unique identifier for that folder. You can modify either the `volume ID` and `directory ID` or just modify the `directory alias` property to set the folder reference. By modifying this property the category is automatically turned into a folder category.

directory ID: For folder categories only: This is the ID of the directory that the folder category refers to. Together with the `volume ID` above this is a unique identifier for that folder. By modifying this property the category is automatically turned into a folder category.

directory: For folder categories only: This is an alias to the folder that the category refers to. This property can only be read if the corresponding volume is mounted. This property is very useful if you want to set the folder reference to an existing folder on one of your mounted disks. By modifying this property the category is automatically turned into a folder category.

table

Description

This object describes a table of a Cumulus database. Each Cumulus database contains two tables: table index 1 is named "Catalog" and contains all catalog records. Table index 2 named "Categories" contains all the categories of the database. You can access all elements of the catalog table object directly from the corresponding database object.

Elements

row: The records of the table.

column: The data fields of the table.

cell: The cells of all rows and all columns.

Properties

properties: A record containing all properties of the given object.

access: Cumulus only allows read access to tables.

best type: The best data type for a table object: `record`. This record contains all properties of the table object.

class: The object class for this kind of objects: `table`.

default type: The same as `best type`.

kind: The kind of the table: `table`.

lock: Cumulus does not support locking of tables and therefor always returns no lock.

name: The name of the table. The first table is always called "Catalog", the second is always "Categories".

selection: The currently selected rows of the corresponding table. This is the selection that the user made in either the catalog window (catalog table) or category window (categories table).

Object Classes of the Cumulus Suite

application preferences: This object represents the settings found in the Preferences dialog of Cumulus.

cataloging preferences: This object represents the settings of the Cataloging Options window of Cumulus. You can read and write any of these preferences that affect the way Cumulus catalogs files.

database info: This object contains information about the database file. It consists of the information that can be found in the administration view of the database window in Cumulus including the log entries.

database settings: This object class represents the settings that can be made in the Database Setup window of Cumulus. It also includes the user permissions.

file filter: A file filter as it can be found in the Cataloging Options window. You can activate and deactivate those filters through AppleScript.

image status: An image status as it can be found in the status list of Cumulus.

category: A category of a database as it appears in the category window of Cumulus. Like the folder structure of the Finder Cumulus uses categories that include sub-categories as elements to represent the hierarchical structuring of categories. You can also access the categories through the categories table of the database object.

log entry: This object represents the log entry that can be seen in the administration view of the database window in Cumulus.

user permissions: A user entry of the Database Setup window of Cumulus.

application preferences

Description

This object represents the settings that can be made using the Preferences dialog in Cumulus.

Properties

properties: A record containing all properties of the given object.

open database view (administration view/text view/thumbnail view/as closed): an enumerated value indicating the view to use for opened databases.

open at startup: A boolean indicating whether Cumulus should open the databases automatically that were open when the program was quit.

search method (fast search/extensive search): The method to use when resolving file references: **fast search** does a fast search including user interaction. **extensive search** performs a full-pathname search first, then searches for the file on all currently mounted volumes. If the file cannot be found Cumulus then tries the fast search method.

show record name: Determines whether Cumulus should display the record names underneath the thumbnail in Thumbnail Display.

units (inch/centimeter/millimeter/point/pixel/pica): The measurement unit used for displaying width and height of images.

window to open on double click (information window/preview window): The kind of window that should be opened when double-clicking a record in the catalog window.

show warnings: Determines whether Cumulus should show warning dialogs.

preview size (original image size/fit to screen/other preview size): Describes the way Cumulus determines the size of a preview. If this is set to "other preview size" then it takes the value of the property **manual preview size**.

manual preview size: The size of a preview in pixels on screen. It is only taken if the property **preview size** has the value "other preview size". This describes the maximum square area where the preview should fit into.

preview display (uncalibrated/calibrated/calibrated with gamut alarm): Determines the way a preview is calibrated for display. In order to use "calibrated" and "calibrated with gamut alarm" you need ColorSync version 2.0 or later installed. If this property is set to "calibrated with gamut alarm" the properties **printer profile** and **gamut alarm color** are used to determine which printer is checked and how the pixels that are outside the color range are displayed in the preview.

printer profile: The name of a printer profile or the file of a printer profile for preview calibration with gamut alarm. The profile has to be installed in the system or in the list of preferred printer profiles available in the application preferences dialog of Cumulus.

gamut alarm color: The color that is displayed for pixels that are out of the printer gamut.

search matching all selected categories: Determines whether searching for categories in the categories window matches all selected categories.

search for categories above: Determines whether searching for categories will also search for broader categories above.

search for categories below: Determines whether searching for categories will also search for narrower categories below.

resolve related categories: Determines whether searching for categories will also search for related categories.

thumbnail similarity search threshold: Determines how the similarity search will find images. The default values are {10, 30}. Use smaller values to do a narrower search. A search for {0, 0} should find identical thumbnails.

cataloging preferences

Description

This object represents the settings that can be made in the Cataloging Options window in Cumulus. There is only one instance of this object accessible through the property **cataloging options** of the application object.

Elements

file filter: The installed file filters. You can activate and deactivate certain filters in this list.

Properties

properties: A record containing all properties of the given object.

status: The preset status.

user: The preset user name.

notes: The preset notes.

categories: The preset categories as a list of strings.

include notes from image file: A boolean indicating whether Cumulus should take notes stored in the image file to be cataloged.

include categories from image file: A boolean indicating whether Cumulus should take categories stored in the image file to be cataloged.

include Finder comments from image file: A boolean indicating whether Cumulus should read the Finder comments from the image file into the notes field.

include folder names as categories: A boolean indicating whether Cumulus should add the folder names of the folder the image file is located to the category list.

open record window automatically: A boolean indicating whether Cumulus should automatically open the record information window for each cataloged file.

excluded files: A list of aliases for the files that should not be cataloged.

exclude damaged: A boolean indicating whether Cumulus should add files to the exclude list if they crash when being cataloged.

cataloging action: (add only/add or update/update only) This property indicates whether Cumulus should add files or update files when cataloging through Drag&Drop, the menu command or the AppleScript **catalog** command.

add duplicates: A boolean indicating whether Cumulus should add files that are already cataloged in the corresponding database.

resolve aliases: A boolean indicating whether Cumulus should resolve alias files to catalog the referenced file. If not alias files will be ignored.

copy file to collection folder: A boolean indicating whether Cumulus should copy the file to be cataloged to the collection folder specified in the database setup when cataloging.

show progress bar: A boolean indicating whether Cumulus should count all the files to be cataloged prior to cataloging them. This may take longer but allows Cumulus to show a progress bar instead of just a changing name during the cataloging process.

line art quality: The thumbnail quality for line art images: **normal quality** or **high quality**.

grayscale quality: The thumbnail quality for grayscale images: **normal quality** or **high quality**.

color quality: The thumbnail quality for color images: **normal quality** or **high quality**.

thumbnail source: The source for creating the thumbnail: **stored thumbnail** or **entire image**.

optimizing thumbnails: A boolean indicating whether Cumulus should automatically optimize the thumbnail of the cataloged image file.

category

Description

This object represents a category. Because the categories are organized hierarchical each category may contain other categories (sub-categories). In this sense the category object is similar to the folder object the Finder. The topmost categories are accessible as elements of the corresponding database where the sub-categories can be accessed as elements of their respective parent category. If you want to access all categories regardless of the hierarchical structure you can do so by accessing the rows of the categories table of the database.

When getting a whole category object Cumulus returns its name by default.

Elements

properties: A record containing all properties of the given object.

best type: The best data type representing a category object: **record**. This record contains all properties of the category object.

class: The class for category objects: **category**.

default type: When getting a whole category object Cumulus returns its name so the default type for objects of this class is **string**.

id: The unique ID of this category. This ID is the same as the ID of the corresponding row in the categories table.

name: The name of the category. This is the default value of the object.

collapsed: A boolean indicating whether this category hides its sub-categories.

level: The level of this category in the hierarchy starting with 1 for the topmost categories.

category type: The kind of this category: **normal category**, **related category**, or **folder category**. These category types have additional properties for additional data: related categories offer the ID of the original category to be read and written. Folder categories allow you to read and write the folder reference either by ID or by an alias. See details below.

original category ID: For related categories only: the ID of the original category. Modifying this property of a normal category or folder category turns the category into a related category with the given original category id.

volume ID: For folder categories only: This is the ID of the volume that the folder category refers to. Together with the **directory ID** below this is a unique identifier for that folder. You can modify either

the `volume ID` and `directory ID` or just modify the `directory alias` property to set the folder reference. By modifying this property the category is automatically turned into a folder category.

directory ID: For folder categories only: This is the ID of the directory that the folder category refers to. Together with the `volume ID` above this is a unique identifier for that folder. By modifying this property the category is automatically turned into a folder category.

directory: For folder categories only: This is an alias to the folder that the category refers to. This property can only be read if the corresponding volume is mounted. This property is very useful if you want to set the folder reference to an existing folder on one of your mounted disks. By modifying this property the category is automatically turned into a folder category.

database info

Description

This object stores the information that can be found in the administration view of the database window in Cumulus.

Elements

log entry: The log entries visible in the administration view of the database window in Cumulus. Only the memory-resident log is available. Because Cumulus stores the log in a text file you can read a log file by using a scriptable text editor. The log entries are ordered in a way that the oldest entry has index 1. Please note that the maximum number of entries is limited by the database settings. If the limit is reached the whole list is scrolled and the oldest entry is removed so the index of each entry changes. Log entries can only be accessed by indexed access.

Properties

properties: A record containing all properties of the given object.

records deleted: The number of records that have been deleted and can be recovered by sending the `recover records` command.

memory free: The number of free bytes in this database. You can reclaim them by sending the `compress` command.

disk space available: The number of bytes free on the disk where the database file is stored.

memory available: The number of free bytes in Cumulus' memory partition.

memory required: The amount of memory that is required to operate on the database. This is the size of the largest index. If this value is larger than the value of the `memory available` property Cumulus may not be able to operate successfully.

memory recommended: The amount of memory that is recommended to operate on the database. This is the total size of all the indexes. Keeping them in memory speeds up searching and scrolling.

database settings

Description

This object represents the settings that can be made in the Database Setup dialog of Cumulus. This object is accessible through the `settings` property of the database object.

Elements

user permissions: the users with the corresponding access permissions for this database. These permissions are only used by Cumulus if the **restricted permissions** property is on.

Properties

properties: A record containing all properties of the given object.

update database file immediately: A boolean indicating whether Cumulus should flush the cached information immediately to the database file.

keep deleted: A boolean indicating whether Cumulus should keep record that are deleted for later recovery.

logging: A boolean indicating whether Cumulus should keep a log of all activities that have changed the database.

log length: The number of entries stored in the memory-resident log. This is the maximum number of objects of class **log entry** that are available as elements of the **database info** object.

save log to file: A boolean indicating whether Cumulus should save the log also to a file. The property **log file** has to contain a valid file specification before you can set this property to **true**.

log file: The file specification of the text file where Cumulus should write the log to. This is only used when the property **save log to file** is **true**.

flush log constantly: A boolean indicating whether Cumulus should update the log file immediately after a new log entry has been added to the memory-resident log. If not Cumulus will only update the log file when closing the database or when the memory-resident log is full.

collection folder usage: An enumerator indicating how Cumulus should use the collection folder specified in the property **collection folder**: **never** means it is not used, **always** means that every file is copied to the folder when it is cataloged, **optional** means that the user can choose with the cataloging option whether he wants to use the collection folder or not. You have to set the property **collection folder** to a valid folder before setting this property to a value other than **never**.

collection folder: An alias specifying a folder to use as the collection folder. see the description of the **collection folder usage** property above.

thumbnail size: A long integer specifying the size for thumbnails of newly created records. Cumulus supports the following values: 96, 128, 192. Other values are mapped to the nearest supported size.

has password: A boolean indicating that the database has a password for write access. The actual password can not be accessed. This property is read-only.

restricted permissions: A boolean indicating whether Cumulus should restrict client access to the database by using the permissions specified in the **user permissions** objects. This property is only available for databases shared by a Cumulus server.

allow sharing: A boolean indicating whether Cumulus should present this database on the network to allow clients to connect to it. This property is only available for databases opened by a Cumulus server.

publish to Internet: A boolean indicating whether Cumulus should present this database on the Internet using the Cumulus Internet Image Server.

file filter

Description

This object represent a file filter installed in the running Cumulus program. You can access these objects as elements of the cataloging preferences object.

Plural Form

file filters

Properties

properties: A record containing all properties of the given object.

name: The internal name of the file filter as it appears in the file filter list of the Cataloging Options dialog in Cumulus.

file type: The file type that this file filter supports.

active: A boolean indicating whether this file filter is currently active.

image status

Description

This object represents a single image status of the status list of a database. These objects can be accessed as elements of the database object.

Elements

properties: A record containing all properties of the given object.

best type: The best data type representing a image status object: **record**. This record contains all properties of the image status object.

class: The class for image status objects: **image status**.

default type: When getting a whole image status object Cumulus returns its name so the default type for objects of this class is **string**.

id: The unique ID of this image status.

name: The name of the image status . This is the default value of the object.

log entry

Description

This object represents a single entry in the memory-resident log of a Cumulus database. Objects of this class can be accessed as elements of the **database info** object and are read-only.

Plural Form

log entries

Properties

properties: A record containing all properties of the given object.

name: The name of the user

log date: The date the action occurred

log action: The kind of action that happened

log object: An additional name of the object that was involved.

user permissions

Description

This object specifies the access permissions for a client that is connecting to the database over a network. This object can be found as elements of the **database settings** object.

Properties

properties: A record containing all properties of the given object.

name: The name of the client user as it appears in the Users & Groups Control Panel.

open database: A boolean indicating whether the user is given the right to open this database.

modify categories: A boolean indicating whether the user is given the right to change the categories in this database.

modify status: A boolean indicating whether the user is given the right to change the status list of this database.

modify records: A boolean indicating whether the user is given the right to change the records of this database.

transfer files: A boolean indicating whether the user is given the right to transfer files from the Cumulus server to the client computer through the client-server connection.

delete records : A boolean indicating whether the user has the permission to delete records from this database.

Commands

Cumulus supports a variety of commands that can sometimes be used on different kind of objects.

The commands are separated into different sets (called "Suites") to build groups that can be used by many different applications. Cumulus supports commands from the predefined "Required Suite", "Standard Suite", and "Database Suite". It also supports commands specific to Cumulus that allow maintenance and retrieval of Cumulus databases.

Commands of the Standard Suite

The "Standard Suite" is a set of commands defined by Apple that should be supported by most applications. These are general commands for manipulating data directly and some standard behavior of applications.

close

Parameters

`reference`: the window or database object to close

`saving` `yes/no/ask` (optional): an enumerated value determining the way the closing is handled. The default is "ask".

Description

Close the given database or window. Closing a database implies closing of all related windows (except preview windows). The `savings` option is used for each of the related closings. When specifying a `savings` option the user may be asked whether he wants changed to be saved. This option only has effect when closing a record information window.

Example

```
close front database saving yes
```

See Also

`open`

count

Parameters

reference: the container object whose elements should be counted

each type class: the type class of elements to count in the container object

Description

Count the number of elements of the given class inside the given container object.

Result

integer: the number of elements

Example

```
count each database
```

Count the number of open databases.

```
set numRecords to count each row of table "Catalog" of front database
```

Set a variable to the number of records currently shown in the frontmost database

```
count categories of first row of selection
```

Return the number of categories assigned to the first selected record.

See Also

`exists`

data size

Parameters

`reference`: the object to get the size
`as type class` (optional): The desired type for the data.

Returns

`integer`: the number of bytes of the data

Description

Get the data size of an object. This is similar to getting the data but only measures the size that would be returned by a corresponding get command.

Example

`data size of name of front database`
Returns the number of bytes (characters) of the database name.

See Also

`get`

delete

Parameters

`reference`: the object to delete. This can be any object of the following classes: row, category, image status, user permissions. It can also be the selection - either in a given window or in the application.

Description

Permanently delete the given objects. Deleting the selection deletes each elements (row) contained in the selection.

Example

`delete user permissions "Mike Ruthless" of settings of front database`

See Also

`make`, `count`

duplicate

Parameters

`reference`: the object to duplicate. Cumulus supports duplication of rows and categories.
`to location reference`: the destination location for the object.

Returns

`reference`: The reference of the object after it has been duplicated.

Description

Duplicate the given objects to the new location. This is very similar to the move command except that the original object is preserved. Using this command you can copy rows, category objects or user permissions from one database to the other.

Cumulus can duplicate category objects to another location in the category list of any database. When moving categories to another database the categories are added to the database at the given location.

You can also duplicate rows of any table to any other table of the same kind. When duplicating categories table rows you can determine the location where the categories are inserted.

Example

```
duplicate every category of database 1 to end of database 2
```

Copy all categories of one database to the other database.

See Also

make, move

exists

Parameters

reference: the object specifier of the object in question

Returns

boolean: true if the object exists, false if not

Description

Test whether the given reference is a valid reference of an existing object.

Example

```
if (exists row 1 of selection) then
    delete row 1 of selection
end if
```

Delete the first row of the selection if any is selected

See Also

count

get

Parameters

reference: the object to get

as type class (optional): The desired type for the returned data.

Returns

anything: the data of the object in the desired type

Description

Get the data of the given object. You can optionally specify the type class for the returned data. Each object (described above) has a default type and a best type to return its data. The best type is the type class that contains the most information about the object. The default type is the data type that is returned if you do not specify an **as** option.

AppleScript automatically issues a **get** command whenever you use the object instead of a direct value in your script.

Example

```
get file reference of row 1 of front database as string
```

Return the file reference of the first row as a full pathname.

```
set dbName to name of front database
```

Get the name of the front database and set a local variable to it. This is equivalent to the following script:

```
set dbName to (get name of front database)
```

See Also

set

make

Parameters

new type class: the class of element to create. Cumulus supports to create elements of the following classes: row, category, image status, user permissions

at location reference: The location at which to insert the new element

with data anything (optional): the initial data for the new element

with properties record (optional): the initial properties for the new element

Returns

reference: A reference to the newly created object

Description

Create a new element of the given class in the given container object. You can supply initial data or properties with the command.

The **at** parameter specifies the location and the container object at which to create the new element.

When creating a new row in a catalog table the location is ignored and only the container object (either a database or a table object) is used.

When creating a new row in the categories table the location is also ignored and the container is the categories table of a database.

When creating a new category the location is also ignored and the container object is either the database or the parent category.

Example

```
make new row at end of front database with properties {file reference: alias  
"Macintosh HD:image file", notes: "Comments on the image file"}
```

Creates a new catalog entry setting some of the record values. The other values are initialized by Cumulus.

```
make new category at end of category "Animal" of front database with data "Mouse"
```

This creates a new sub-category of the topmost category called "Animal" giving the name "Mouse"

```
make new user permissions at after user permissions "Edward" of settings of front  
database with data "Francis"
```

This adds a new user to the user list and places it at a special position.

See Also

delete, count

move

Parameters

reference: the object to move. Cumulus supports moving rows and categories.

to location reference: the destination location for the object.

Returns

reference: The reference of the object after it has been moved.

Description

Move the given objects to the new location. Move is very similar to the duplicate command except that the original object is deleted.

Cumulus can move category objects to another location in the category list of any database. When moving categories to another database the categories are added to the database at the given location just like Drag&Drop works.

You can also move rows of any table to any other table of the same kind. When moving categories table rows you can determine the location where the categories are inserted.

Example

```
tell front database
  move category "Mouse" of category "Animal" to end of category "Mammal" of
  category "Animal"
end tell
```

See Also

make, duplicate, delete

open

Parameters

alias: an alias to a database file to be opened or to a file or folder to be cataloged.

with password string (optional): The password for write access to a database file. If you do not specify the password and the database requires such a password you will be asked to type it in. If you have turned off user interaction the database is opened as read-only.

Description

This command is identical to the user action of dragging and dropping a file or folder to the Cumulus application icon in the Finder. If the file is a database file it is opened (Cumulus Desktop PLUS only). Otherwise the file or folder is cataloged to the frontmost database.

Example

```
open alias "Macintosh HD:Cumulus:Image Database"
```

See Also

catalog, open remote database

print

Parameters

Description

Example

See Also

quit

Parameters

saving *yes/no/ask* (optional): an enumerated value determining the way the closing of windows is handled. The default is "ask".

Description

Quit the application. You can specify whether you want to save any changes made to any of the open record information windows.

Example

```
quit saving yes
```

Quits Cumulus saving each window contents without asking.

See Also

`close`

save

Parameters

Description

Example

See Also

set

Parameters

reference: The object to change

to *anything*: the new value

Description

Set the given object to the given value.

Example

```
set file server zone name of every row of selection to "EtherZone"
```

Changes the file server zone name only for the selected catalog entries.

See Also

`get`

Commands of the Database Suite

The commands of the "Database Suite" are defined by Apple and apply to all database oriented programs. This is an extension of the "Table Suite" to allow multiple tables per database. Cumulus does support only the **sort** command to set the sorting of the records in the catalog table.

sort

Parameters

table: The catalog table to sort.

by column (optional): The column of the catalog table be used for sorting

in order *ascending/descending* (optional): The sort order.

Description

Set the sorting of the catalog window. You can change the sorting column and the sort order independently or all together. The following columns can be used for sorting: "Record Name", "File Type", "Color Mode", "Horizontal Resolution", "Vertical Resolution", "Record Modification Date".

The meaning of this command differs from the definition made for the Database Suite. After a search Cumulus sorts all found records in the background and enables the sorting buttons in the catalog window as soon as this is done. You can check the **sorting rows** property of the database to see whether Cumulus is still sorting the records. If you call the **sort** command without checking the property Cumulus will wait until all record have been processed.

Example

```
tell first database
  repeat while sorting rows
    end repeat
  sort table "Catalog" by column "Record Name" in order ascending
end tell
```

Set the sorting of the frontmost database to "Record Name" in ascending order. By using a loop to wait until the sorting is complete you can cancel execution of the script.

Commands of the Cumulus Suite

The Cumulus Suite is to offer all commands that are not included in the standard suites defined by Apple.

add database file

Parameters

string: the path name of a database file.

on string (optional): the Cumulus server to add the database file to.

as string: The user name of the Cumulus administrator.

login password string (optional): The user password of the Cumulus administrator (if any).

database password string (optional): The password for write access to the Cumulus database file (if any).

Returns

string: The name of the database added.

Description

Adds a database file to the list of open files at a Cumulus server. The file is specified as a path name of an existing file. This command returns the name of the database to be used in a subsequent “open remote database” command.

You specify the Cumulus administrator’s user name and password to allow Cumulus to connect to the server temporarily. You can also specify the password for write access to allow the server to open the file with write permissions.

Example

```
add database file "D:\\catalogs\\Media Catalog.ccf" on "TCP/IP:server" as "Peter"  
login password "secret"
```

Logon to the Cumulus server called “server” and add the database called “Media Catalog”. The Cumulus administrator has the user name “Peter” and the password “secret”.

See Also

`choose server, close database file, new database file`

backup

Parameters

`database`: the database to back up

`to file specification`: the file to write the backup copy to.

`full backup` `boolean` (optional): Determines whether you want a minimal backup file (default) or a full backup.

Description

Make a backup copy of the given database. The database is a direct database. This command will first disconnect any clients for this database. Previously deleted records are not stored in the backup copy.

The file specification is either a Mac OS file or a path name local to the Cumulus server when controlling a database of a Cumulus server.

When controlling a remote database you have to connect to the Cumulus server as the Cumulus administrator.

Example

```
backup front database to file "E:\\backups\\backup.ccf" full backup yes
```

See Also

`compress`

check

Parameters

`reference`: the file reference of the rows to check.

Description

Check the given file references and update them accordingly. The parameter has to be the property **file reference** of any row of the catalog table.

Example

```
check file reference of every row of front window
```

See Also

`update`

choose remote database

Returns

`record`: A record structure containing the machine of the Cumulus server, the login information, and the name of the database chosen.

Description

Bring up a dialog to choose a database that is shared by a Cumulus server somewhere in the network.

The result is a record that contains two members: `database server`: the machine where the server is located and `name`: the name of the database. This command brings Cumulus to front temporarily.

Example

```
set db to (choose remote database)
open remote database (name of db) on (database server of db) as (user of db) login
password (login password of db)
```

Bring up a browser dialog to choose a remote database and open the returned database specification.

See Also

`choose server`

choose server

Parameters

`on string` (optional): The name of the protocol to use. Default is "TCP/IP".

Returns

`record`: A record structure containing the machine of the Cumulus server and the user name and password used to connect to the server.

Description

Bring up a connection dialog to choose a Cumulus server in your network. This can be used to let the user determine the Cumulus server that the script is working with.

The result is a record that contains the following members: `database server`: the machine where the server is located `user`: the login name used to connect to the server, `login password`: the password used to connect to the server. This command brings Cumulus to front temporarily.

Example

```
set theServer to (choose server)
add database file "D:\\catalogs\\Media Catalog.ccf" on (database server of theServer)
as (user of theServer) login password (login password of theServer)
```

Bring up a browser dialog to choose a Cumulus server.

See Also

`choose remote database`, `add database file`, `close database file`, `new database file`

close database file

Parameters

string: The name of the database to close. This is the custom database name as visible in the Catalog Setup dialog.

on string: The name of the server in the form <<protocol>>:<<server>>[:<<port>>]. For example "TCP/IP:server".

as string (optional): The user name to use for logging on to the Cumulus server. This has to be the Cumulus Administrator's name.

login password string (optional): The user password for logging on to the Cumulus server. This has to be the Cumulus Administrator's logon password.

Description

Close the given database file at the Cumulus server.

Cumulus connects to the server using the given name and password and instructs the server to close the given database. All clients that are connected to the database will be disconnected.

Example

```
close database file "Media Catalog" on "TCP/IP:server" as "Peter" login password "secret"
```

Logon to the Cumulus server called "server" and close the database called "Media Catalog". The Cumulus administrator has the user name "Peter" and the password "secret".

See Also

choose server, choose remote database, add database file, new database file

compress

Parameters

database: the database to compress

Description

Compress the given database to minimize the file size. The database is a direct database. This command will first disconnect any clients for this database. All previously deleted records are permanently removed from the database and cannot be recovered anymore.

When controlling a remote database you have to connect to the Cumulus server as the Cumulus administrator.

Example

```
compress front database
```

See Also

backup

copy image file

Parameters

`row`: the rows of the catalog table whose image files should be copied.

Description

Copy the image files referenced by the given catalog table rows to a new folder.

Example

```
copy image file row 1 3 to alias "Macintosh HD:new folder:"
```

See Also

move image file, check

catalog

Parameters

`alias`: the files to catalog, can be a list

`to database`: the database to add the files to

Returns

A list of row references for the rows created

Description

Catalog all files specified by the direct parameter to the given database. The alias can either point to a file, a folder or a disk. This command is similar to the **open** command but allows you to specify a database to catalog the files to.

Example

```
catalog alias "Macintosh HD:" to front database
```

See Also

open

export

Parameters

`row`: the rows to export. These can be either rows of the catalog or the categories table.

`to file specification`: The file to write the export data to.

Description

Export the given rows to the file specified. Depending on the table of the rows it is either a catalog record or a categories export.

Example

```
export every row of selection of front window
```

See Also

import

find

Parameters

row: The rows of a catalog table to find

matching *string* (optional): The find query to use.

in *all records/shown records* (optional): The source for the searching.

Returns

The number of records found

Description

Do a search in the catalog table and show only the rows found. There are different ways of specifying the search query. You can either restrict the rows by using the AppleScript "where" clause of AppleScript or you can specify a textual search query. The format of the textual query is the same as the export format for search queries.

Using the "where" clause of AppleScript is easy to type and read but has a fixed structure. Only the values to compare to are variable. If you intend to search using a variable number of conditions or variable comparison operators you should use the **matching** option and create the string on your own.

The **find** command only supports the searches that can be made using the Find window of Cumulus. If your "where" clause is too complex to be translated to the internal query format Cumulus will return an error.

You can not combine the "where" clause query specification and the **matching** option in one **find** command.

Using the **in** option allows you to search in the records that are currently visible. This corresponds to the "Find Again" button in the Find window of Cumulus. The default is to search in all records.

Please note that searching by date takes the day of the given date if you do not specify a time. This behavior is identical to the search you can do in the Find window in Cumulus which also searches for a matching date by ignoring the time.

Examples

To show all records of the database use the following command (this is equivalent to the "Find All" menu command in Cumulus):

```
find every row of front database
```

To perform the search specified by the Find window that is in front do the following (this is equivalent to pressing the "Find" button in the Find window in Cumulus):

```
find every row of database of front window matching search query of front window
```

To show all TIFF files where the record name starts with "Rose"

```
find every row of front database where file type is "TIFF" and name starts with "Rose"
```

You can also search for the same records by using the **matching** option below. Please note that the query format as text uses tab to separate the fields of the query and return to separate the conditions.

```
find every row of front database matching "file type" & tab & "is" & tab & "TIFF" & return & "and" & tab & "record name" & tab & "starts with" & tab & "Rose"
```

You can also write the string with the tab and return typed in but be careful not to add any tab at the beginning of the next line. When you type the return character the Script Editor automatically indents the following line using a tab.

```
find every row of front database matching "file type ->| is ->| TIFF
```

```
and ->| name ->| starts with ->| Rose"
```

The following example does the same in two commands using the **in** option:

```
find every row of front database where file type is "TIFF"
```

```
find every row of front database where name starts with "Rose" in shown records
```

See Also

count, get

import

Parameters

alias: the text file to read records or categories from.

to table: the table where the records will be added to. The kind of table determines whether you want to import catalog records or categories.

Return Value

row: a list of row references for the newly created rows.

Description

Import records or categories from a text file. The text file must be of the format created by the **export** command. Use the table "Catalog" to import catalog records. Use the table "Categories" to import categories.

Example

To import previously exported records:

```
import alias "Macintosh HD:DB Export" to table "Catalog" of front database
```

To import previously exported categories:

```
import alias "Macintosh HD:Category Export" to table "Categories" of front database
```

See Also

export

move image file

Parameters

row: the rows of the catalog table whose image files should be moved.

Description

Move the image files referenced by the given catalog table rows to a new folder. Update the file references to point to the new location.

Example

```
move image file row 1 3 to alias "Macintosh HD:new folder:"
```

See Also

copy image file, check

new database file

Parameters

string: The full path name of the database file to open. This has to be a path that is valid on the target server computer.

on string: the machine the Cumulus server is running on. This is usually specified as a string of the form <<protocol>>:<<server>>[<<port>>] that specifies the protocol ("TCP/IP"), the server (an IP address) and an optional TCP/IP port number.

as string (optional): The user name of the Cumulus administrator for connecting to the server.

login password string (optional): The user password of the Cumulus administrator to connect to the Cumulus server.

Description

Create a new database file at the Cumulus server. You have to connect using the Cumulus administrator's user name and password.

The file name has to be given in the target computer's namespace. For example it has to be a Windows path name if the Cumulus server runs on a Windows NT computer.

After creating the file the Cumulus server will automatically open the database file.

Example

```
new database file "C:\\catalogs\\media.ccf" on "TCP/IP:SERVER"
```

Remember that you have to type two backslash characters to specify a single character for the AppleScript string.

See Also

`choose server`, `open remote database`, `close remote database`

open remote database

Parameters

string: The name of the database.

on machine: the machine the Cumulus server is running on. This is usually specified as a string of the form <<protocol>>:<<server>>[<<port>>] that specifies the protocol ("TCP/IP"), the server (an IP address) and an optional TCP/IP port number.

as string (optional): The user name for connecting to the server. If no user name is specified Cumulus will connect as guest.

login password string (optional): The user password to connect to the Cumulus server.

Description

Open a database that is shared by a Cumulus server in the network. You will have to type in your name and password to connect to the server machine. You can use the command "choose remote database" to let the user select a remote database to open.

Example

```
open remote database "Media Catalog" on "TCP/IP:SERVER"
```

Open the database named "Media Catalog" by connecting to the Cumulus server named "SERVER" using the default TCP/IP port.

```
open remote database "Media Catalog" on "TCP/IP:SERVER" as "Peter" login password "secret"
```

Open the same database but connecting to the server as the user "Peter" using the login password "secret".

```
if remote database of first database then
    set mach to database server of first database
    set db to name of first database
end if
```

```
.
.
.
```

```
open remote database db on mach
```

Remember the remote database specification in local variables and open the same remote database later.

See Also

choose remote database, database info object

optimize

Parameters

reference: the thumbnails of the rows to be optimized.

Description

Optimize the given thumbnails by increasing its contrast and adjusting the brightness. The parameter is the thumbnail property of the rows to optimize.

Example

```
optimize thumbnail of every row of selection
```

See Also

preview

Parameters

row: the rows of the catalog table whose preview will be opened.

Description

Open the preview window of the given rows of the catalog table. Bring the corresponding window to front if it is already open.

Example

```
preview every row of selection of front window
```

rebuild

Parameters

database: the database to rebuild

Description

Rebuild all index information for the given database. The database is a direct database. This command will first disconnect any clients for this database.

When controlling a remote database you have to connect to the Cumulus server as the Cumulus administrator.

Example

```
rebuild front database
```

See Also

backup

recover deleted records

Parameters

database: the database to act on

Description

Recover all deleted records of the given database and show them in the catalog window. To keep deleted records recoverable set the corresponding property of the **database settings** object.

When controlling a remote database you have to connect to the Cumulus server as the Cumulus administrator.

Example

```
recover deleted records front database
```

See Also

backup

rotate

Parameters

`reference`: the thumbnails of the rows to be rotated.

`by` (optional): the angle for the rotation. This has to be a multiple of 90°. Positive values rotate clockwise, negative values counter-clockwise.

Description

Rotate the given thumbnails by the given angle. The direct parameter is the thumbnail property of the rows to rotate.

Example

```
rotate thumbnail of every row of selection of front window by 90
```

Rotate the selected thumbnails clockwise.

select

Parameters

`reference`: the objects to select.

Description

Select any set of objects in the front window. The front window must be a catalog or category window. The objects to select must be rows of the corresponding table or category objects for the category window.

Example

```
select every row of database of front window
```

See Also

`selection object`

show find window

Parameters

`database`: The database whose find window should be opened.

Returns

`window`: The find window.

Description

Open the find window of the given database and bring it to front.

Example

```
search query of (show find window front database)
```

Get the search query of the find window of the frontmost database

See Also

`close`

show information window

Parameters

`row`: The row of a catalog table to act on.

Result

`window`: The record information window opened.

Description

Open the record information window for the given rows of the catalog table. Bring the window to front if it is already open.

Example

```
show information window every row of selection
```

Opens the information window for each selected record.

See Also

`close`

show categories window

Parameters

`database`: The database whose categories window should be opened.

Returns

`window`: The categories window.

Description

Open the categories window of the given database and bring it to front.

Example

```
show categories window of front database
```

See Also

`close`

update

Parameters

`row`: the rows of the catalog table to update

force `boolean` (optional): force an update of the record even if the file modification date has not changed.

Description

Update the given rows of the catalog table. This also updates the file reference stored in the record.

Example

```
update every row of selection of front window
```

See Also

`check`