Jovis Architectural Commands

Syntax Reference

AllMarkerStatus

SYNTAX

```
Get Jovis (
[1] "AllMarkerStatus", ¬
[2] "FileGlobal", ¬
[3] '[Requested Info]' )
```

WHERE

[1] is the Jovis 'AllMarkerStatus' command;
[2] is the global identifier for the file you want to work with;
[3] is any combination of the letters "INTPK", the default is "INTPK"

EXAMPLE

```
on DemoScript
  put Jovis("AllMarkerStatus","myDB","NTK") into field  "Marker List"
end DemoScript
```

DESCRIPTION

Returns information about each marker in the database. Parameter 2 is a string of up to 5 characters indicating the information to return, as follows:

I - Marker ID
N - Marker Name
T - True if marker is temporary,
false if it is not.
P - Key which points to this marker
K - Key which this marker points to

Each item will be separated by a comma and each set of marker information will be separated by a return character.

You do not need to include all 5 characters; only include the characters for the information you need. Currently not available in the client/server version.

COMMENTS

Currently not available in the multi-user version.

SEE ALSO

SetMarker
GoToMarker

BlobToClip

SYNTAX

```
Get Jovis (
[1] "BlobToClip", ¬
[2] "BlobGlobal", ¬
[3] "ResType" )
```

WHERE

[1] is the Jovis 'BlobToClip' command;
[2] is the global identifier for the file you want to work with;
[3] is the resource type.

EXAMPLE

```
on DemoScript
  get Jovis("BlobToClip","BlobGlobal","TEXT")
end DemoScript
```

DESCRIPTION

Copies a 'Blob' to the clipboard and assigns a resource type as well.

SEE ALSO

ReadBlob
WriteBlob
FileToBlob

BlobToResName

SYNTAX

```
Get Jovis (
[1] "BlobToResName", ¬
[2] "BlobGlobal", ¬
[3] "ResType", ¬
[4] "ResourceName", ¬
[5] '[Overwrite Flag]', ¬
[6] '[FilePath]' )
```

WHERE

[1] is the Jovis 'BlobToResName' command;
[2] is the global identifier for the Blob you want to work with;
[3] is the resource type to create or use;
[4] is the resource name to be used;
[5] is an optional parameter. If a resource of the given type and name already exists and this parameter is "true", the resource is overwritten. Otherwise, an error is returned.
[6] is an entire file pathname to an external file. If this parameter is empty or missing the current resource fork is used.

EXAMPLE

```
on DemoScript
  get Jovis("BlobToResName","myDB","ICON","Opus","true")
end DemoScript
```

DESCRIPTION

Takes a 'Blob' global variable and writes it to either the current resource fork, or the resource fork of a given file using the resource type and name provided.

BlobToResID

SYNTAX

```
Get Jovis (
[1] "BlobToResID", ¬
[2] "BlobGlobal", ¬
[3] "ResType", ¬
[4] "ResourceID", ¬
[5] '[ResourceName]', ¬
[6] '[Overwrite Flag]', ¬
[7] '[File Path]' )
```

WHERE

[1] is the Jovis 'BlobToResID' command;
[2] is the global identifier for the file you want to work with;
[3] is the resource type to create or use;
[4] is the resource ID to use;
[5] is an optional name to use for the resource;
[6] is an optional parameter. If a resource of the given type and name already exists and this parameter is "true", the resource is overwritten. Otherwise, an error is returned.
[7] is an entire file pathname to an external file. If this parameter is empty or missing the current resource fork is used.

EXAMPLE

```
on DemoScript
  get Jovis("BlobToResID","myDB","ICON","1391","Opus","true")
end DemoScript
```

DESCRIPTION

Takes a 'Blob' global variable and writes it to either the current resource fork, or the resource fork of a given file using the resource type and ID provided.

ChangeDBPassword

SYNTAX

```
Get Jovis (
[1] "ChangeDBPassword", ¬
[2] "FileGlobal", ¬
[3] "Read-Write Password", ¬
[4] "New Password", ¬
[5] '[Read-Only Flag]' )
```

WHERE

[1] is the Jovis 'ChangeDBPassword' command;
[2] is the global identifier for the file you want to work with;
[3] is the current Read-Write password;
[4] is the new password to be installed;
[5] is the Read-Only flag. If readOnly flag is true, then the password being changed is the readOnly password. If the flag is false or empty, then the password being changed is the read/write password.

EXAMPLE

```
on DemoScript
  put Jovis("changeDBPassword","myDB","Clark","Lois",true) into temp
end DemoScript
```

DESCRIPTION

Changes a current password for either the 'Relational' or 'Architectural' data file. The second parameter must be the current read/write password for changing either the read-write or read-only passwords.

COMMENTS

In the multi-user version, any client trying to sign on to the database after the password is changed must use the new password.

SEE ALSO

CreateCollection
DBInit

ClearBlob

SYNTAX

```
Get Jovis (
[1] "ClearBlob", ¬
[2] "BlobGlobal" )
```

WHERE

[1] is the Jovis 'ClearBlob' command;
[2] is the global identifier for the Blob you want to work with;

EXAMPLE

```
on DemoScript
  global BlobGlo
  get Jovis("ClearBlob","BlobGlo")
  put empty into BlobGlo
end DemoScript
```

DESCRIPTION

Clears the Blob global variable and releases the memory used by it.

ClearCurRec

SYNTAX

```
Get Jovis (
[1] "ClearCurRec", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'ClearCurRec' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  get Jovis("ClearCurRec","myDB")
end DemoScript
```

DESCRIPTION

Clears the current record of it's status as the current record. This command does not touch the current key or the record pointed to by the current key.

SEE ALSO

Currency Tutorial

ClearPict

SYNTAX

```
Get Jovis (
[1] "ClearPict", ¬
[2] "PictGlobal" )
```

WHERE

[1] is the Jovis 'ClearPict' command;
[2] is the global identifier for the 'Pict' you want to work with;

EXAMPLE

```
on DemoScript
  get Jovis("ClearPict","PictVar")
end DemoScript
```

DESCRIPTION

Clears the Pict global variable and releases the memory used by it.

ClearSound

SYNTAX

```
Get Jovis (
[1] "ClearSound", ¬
[2] "SndGlobal" )
```

WHERE

[1] is the Jovis 'ClearSound' command;
[2] is the global identifier for the 'snd ' you want to work with;

EXAMPLE

```
on DemoScript
  get Jovis("ClearSound","soundVar")
end DemoScript
```

DESCRIPTION

Clears the 'snd ' global variable and releases the memory used by it.

ClipToPict

SYNTAX

```
Get Jovis (
[1] "ClipToPict", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'ClipToPict' command;
[2] is the global identifier for the 'Pict' you want to work with;

EXAMPLE

```
on DemoScript
  global pictVar
  put "Jovis" into pictVar
  get Jovis("ClipToPict","pictVar")
end DemoScript
```

DESCRIPTION

Copies the picture currently stored in the clipboard into the 'Pict' Variable.

ClipToSound

SYNTAX

```
Get Jovis (
[1] "ClipToSound", ¬
[2] "SndGlobal" )
```

WHERE

[1] is the Jovis 'ClipToSound' command;
[2] is the global identifier for the ' snd' you want to work with;

EXAMPLE

```
on DemoScript
  global SndGlobal
  if SndGlobal is empty then
    put "Jovis" into SndGlobal
    get Jovis("ClipToSound","SndGlobal")
  end if
end DemoScript
```

DESCRIPTION

Copies the 'snd ' currently stored in the clipboard into the 'Pict' Variable.

ClipToPictFile

SYNTAX

```
Get Jovis (
[1] "ClipToPictFile"
[2] "File PathName", ¬
[3] '[CreatorType]' )
```

WHERE

[1] is the Jovis 'ClipToPictFile' command;
[2] is the entire file pathname for creating a pict file;
[3] is an optional four charactor creator type.

EXAMPLE

```
on DemoScript
  get Jovis("ClipToPictFile","Hard Disk:Pictures Folder","8BIM")  -->
         Adove Photoshop creator
end DemoScript
```

DESCRIPTION

This utility command takes a PICT located on the clipboard and turns it into a PICT file with the given name. It does not matter whether the PICT came from the shell application's paint tools, from another XCMD, or from another application altogether, or from Jovis.

The optional parameter CreatorType is a four-character string which will be the creator type of the file. This will allow the file to be double-clicked to invoke the corresponding application. The default value is '????' -- which means no application.

CopyPict

SYNTAX

```
Get Jovis (
[1] "CopyPict", ¬
[2] "PictGlobal", ¬
[3] "Rect Coordinates" )
```

WHERE

[1] is the Jovis 'CopyPict' command;
[2] is the global identifier for the 'Pict' you want to work with;
[3] is a four item string of coordinates.

EXAMPLE

```
on DemoScript
  global thePict
  put "Jovis" into thePict
  get Jovis("copyPict","thePict","30,40,75,100")
end DemoScript
```

DESCRIPTION

Copies the picture enclosed by the designated rectangle.

SEE ALSO

Initializing a Jovis Global

CountKeys

SYNTAX

```
Put Jovis (
[1] "CountKeys", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'CountKeys' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  put Jovis("countKeys","myDB") into numberOfKeys
end DemoScript
```

DESCRIPTION

Returns the number of keys in the current keyset.

DBClose

SYNTAX

```
Get Jovis (
[1] "DBClose", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'DBClose' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  get Jovis("DBClose","myDB")
end DemoScript
```

DESCRIPTION

Closes the database file.

COMMENTS

Don't forget to call 'ShutDown' before quitting the shell application.

SEE ALSO

ShutDown

DBInit

SYNTAX

```
Get Jovis (
[1] "DBInit", ¬
[2] "myDB", ¬
[3] '[File Name]', ¬
[4] '[Prompt]', ¬
[5] '[Error Global]', ¬
[6] '[Creator',[File Types]]', ¬
[7] '[Dialog Coordinates]', ¬
[8] '[Read-Only PW]', ¬
[9] '[Read-Write PW]' )
```

WHERE

[1] is the Jovis 'DBInit' command;
[2] is the global identifier for the file you want to work with;
[3] is the entire pathname to re-initilize or create a new data file. If this parameter is empty, the standard file put dialog is displayed;
[4] is the prompt for the standard file put dialog;
[5] is is the name of a global variable in which you want Jovis to return error codes. All error codes are returned in this global rather than as part of a function's returned value.
[6] is the optional creator and file types to be assigned to the data file. For example: "DASW,PROR" — which are also the defaults;
[7] is the coordinates for positioning the standard file put dialog;
[8] is the Read-Only' password to be assigned to the data file;
[9] is the Read-Write' password to be assigned to the data file.

EXAMPLE

```
on DemoScript
  global myData
  put "Jovis" into myData
  get Jovis("DBInit","myData")
  get Jovis("DBInit","myDB","HD20:Data:Test","","DBErrors")
  get Jovis("DBInit","myDB",empty,"Which file?")
  get Jovis("DBInit","myDB",empty,"Name this
         beast!","110,110","DBFL","secret","super secret")
  -- Client/Server:
  get Jovis("DBInit","myDB","Data:Testserver@TwilightZone","","DBErrors")
end DemoScript
```


DESCRIPTION

Initializes a hierarchical database file. If the file does not exist, it creates a new one. (In the multi-user version, DBInit does not initialize an existing file.)

COMMENTS

The Jovis Global must be initialized to “Jovis” before using this command. DBInit changes the global to a global identifier.

All errors are reported using th "JovisErrorCode" global.

For the multi-user version:
If the named file does not exist, DBInit creates a new file in the same folder in which the application Jovis Server is located. If the named file already exists, an error message is returned. The status of the newly-created database is open exclusive to the client that created it. This allows the client the opportunity to define the database's internal structures. As soon as the client closes the file, it becomes available to other users on the network.

In the multi-user version, pathname is called entity name. The syntax for an entity name; is:

object name:server name@zone name

For example,
Customers:Jovis_Server@CustomerService

Do not put any spaces between the file name and the colon, or the Server name and the "@".

Each name (object name, server name, zone name) is limited to 31 characters. If there are no zones on your network, or if the Server is in the same zone as the client, use “*” for the zone name.

SEE ALSO

Initializing a Jovis Global
Exclusive Status

DBOpen

SYNTAX

```
Get Jovis (
[1] "DBOpen", ¬
[2] "myDB", ¬
[3] '[File PathName]', ¬
[4] '[Prompt]', ¬
[5] '[Read-Only Flag]', ¬
[6] '[Error Global]', ¬
[7] '[Password]', ¬
[8] "Exclusive Flag]", ¬
[9] '[Dialog Coordinates]' )
```

WHERE

[1] is the Jovis 'DBOpen' command;
[2] is the global identifier for the file you want to work with;
[3] is the entire pathname to a data file to open. If this parameter is empty, the standard file get dialog is displayed;
[4] is the prompt for the standard file get dialog;
[5] is the "True" or "False" flag for whether the file should be opened read-write (true), or read-only (false). The default is "true"; read-write.
[6] is is the name of a global variable in which you want Jovis to return error codes. All error codes are returned in this global rather than as part of the function's returned value.
[7] is the 'Read-Write' password, if one has been assigned to the file;
[8] the exclusive flag is recognized only in the multi-user version. In the single-user version it is ignored.
[9] is the coordinates for positioning the standard file get dialog.

EXAMPLE

```
on DemoScript
  global myData
  put "Jovis" into myData
  get Jovis("DBOpen","myData")
  get Jovis("DBOpen","myDB","HD20:Data:Test")
  get Jovis("DBOpen","myDB",empty,"Which
         file?",true,,myPassword,false,DlogCoordinates) into temp
  --Client/Server:
  get Jovis("DBInit","myDB","Data:Testserver@TwilightZone","","DBErrors")
end DemoScript
```


DESCRIPTION

Opens an already existing hierarchical database file.

COMMENTS

All errors are reported using the "JovisErrorCode" global.

Variations for multi-user version:

If the file name is empty, then the dialog box to open files on the network is displayed.

The syntax for an entity name; is:

object name:server name@.i.zone name

For example,
Customers:Power_Server@CustService

Do not put any spaces between the file name and the colon, or the Server name and the "@". Each name (object name, server name, zone name) is limited to 31 characters.

Networks without zones:
If there are no zones on your network, or if the Server is in the same zone as the client, use “*” for the zone name.

The file must already be opened at the Server before the client can open it.

Using the Exclusive Flag causes the file to be opened for just one client. Other clients are denied access as long as the exclusive flag is set.

SEE ALSO

Initializing a Jovis Global
Exclusive Status

DefineKeyset

SYNTAX

```
Get Jovis (
[1] "DefineKeyset", ¬
[2] "FileGlobal", ¬
[3] "Key size", ¬
[4] "Dup Keys Allowed", ¬
[5] "Key Type", ¬
[6] "Keyset Name" )
```

WHERE

[1] is the Jovis 'DefineKeyset' command;
[2] is the global identifier for the file you want to work with;
[3] is the key size for 'Text' key types. For number keys the size is 10, and for date keys it is 4. Text keys can be from 1 to 128 characters.
[4] is 'True' or 'False' for allowing duplicates keys in the keyset;
[5] is the key type. This affects the ordering of the keys in the keyset.Valid key types are: “N” for numeric, or “D” for date, and any other value (including no parameter) for text.
[6] is the name for the keyset.

EXAMPLE

```
on DemoScript
  get Jovis("DefineKeyset","myDB",6,false,"PictureItem")
end DemoScript
```

DESCRIPTION

Defines a keyset for use within the data file.

COMMENTS

All six parameters are required. Only dates and not times are supported. Numbers with up to 16 significant decimal places will be recognized by Jovis. But remember that computations in your shell application may round your result to only 6 significant decimal places.

SEE ALSO

Valid Names

DelKey

SYNTAX

```
Get Jovis (
[1] "DelKey", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'DelKey' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  get Jovis("DelKey","myData")
end DemoScript
```

DESCRIPTION

Deletes the current key and positions you to the next key in the keyset.

COMMENTS

If there is no key following the current key, delKey positions you on the prior key. If there are no more keys in the keyset, the current key indicator is invalid after delKey completes.

DelKeyset

SYNTAX

```
Get Jovis (
[1] "DelKeyset", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'DelKeyset' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  get Jovis("DelKeyset",et("myData")
end DemoScript
```

DESCRIPTION

Deletes the current keyset and it's keys.

COMMENTS

If the keyset is pointed to directly by a marker, it cannot be deleted or unlinked until the marker is removed.

DelRec

SYNTAX

```
Get Jovis (
[1] "DelRec", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'DelRec' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  get Jovis("DelRec","myData")
end DemoScript
```

DESCRIPTION

Unlinks the current key from its record, and deletes the record.

COMMENTS

If other keys point to the record, the record becomes the current record. If no other keys point to the record, the record is physically deleted from the database.

Variations for multi-user version:

In order to be deleted, the record must have been locked when it was read. DelRec unlocks the record.

DisplayPict

SYNTAX

```
Get Jovis (
[1] "DisplayPict", ¬
[2] "FileGlobal", ¬
[3] "Coordinates", ¬
[4] '[Use Dialog Flag]' )
```

WHERE

[1] is the Jovis 'DisplayPict' command;
[2] is the global identifier for the file you want to work with;
[3] is the location to position the 'Pict';
[4] if missing or false the 'Pict' is drawn on the shell application's window. If "Ture" a dialog window is used.

EXAMPLE

```
on DemoScript
  get DisplayPict("pictVar")
  get DisplayPict("pictVar","355,300")
  get DisplayPict("pictVar","50,50","True")
end DemoScript
```

DESCRIPTION

Draws the picture pointed to by the 'Pict' global onto the screen or into a dialog window.

COMMENTS

The picture is not pasted on the card, but merely drawn over it. This means that the shell application does not automatically redraw the picture if it needs to be redrawn.

FileToBlob

SYNTAX

```
Get Jovis (
[1] "FileToBlob", ¬
[2] "FileGlobal", ¬
[3] "BlobGlobal", ¬
[4] "Entire PathName", ¬
[5] '[Prompt]' )
```

WHERE

[1] is the Jovis 'FileToBlob' command;
[2] is the global identifier for the file you want to work with;
[3] is the global identifier for the 'Blob' you want to work with;
[4] is the file pathname to read the file. If missing or empty the standard file get dialog is displayed;
[5] is the prompt for the standard file get dialog.

EXAMPLE

```
on DemoScript
  global BlobGlobal
  put "Jovis" into BlobGlobal
  get Jovis("FileToBlob","myDB","BlobGlobal","","Select file:")
end DemoScript
```

DESCRIPTION

Reads all information in a file's data fork to a Jovis initialized 'Blob' global.

COMMENTS

Be sure that you have enough memory for the file you are about to bring in.

SEE ALSO

Initializing a Jovis Global
WriteBlob

FileToPict

SYNTAX

```
Get Jovis (
[1] "FileToPict", ¬
[2] "PictGlobal", ¬
[3] "Entire PathName", ¬
[4] '[Prompt]' )
```

WHERE

[1] is the Jovis 'FileToPict' command;
[2] is the global identifier for the 'Pict' you want to work with;
[3] is the file pathname to read the 'Pict'. If missing or empty the standard file get dialog is displayed;
[4] is the prompt for the standard file get dialog.

EXAMPLE

```
on DemoScript
  global PictGlobal
  if PictGlobal is empty then
    put "Jovis" into PictGlobal
    get Jovis("FileToPict","PictGlobal","","Select File:")
  end if
end DemoScript
```

DESCRIPTION

Reads a 'Pict' file to a Jovis initialized global.

SEE ALSO

Initializing a Jovis Global

FindAt

SYNTAX

```
Put Jovis (
[1] "FindAt", ¬
[2] "FileGlobal", ¬
[3] "key" )
```

WHERE

[1] is the Jovis 'FindAt' command;
[2] is the global identifier for the file you want to work with;
[3] is the key to locate within the keyset. Partial key strings can also be used.

EXAMPLE

```
on DemoScript
  put Jovis("FindAt","myDB","Smith") into temp
end DemoScript
```

DESCRIPTION

Finds the first key in the current keyset whose value is greater than or equal to the value passed as parameter 2, and makes it the current key. If there is no exact match, warning 11 is returned. If there are no keys greater than or equal, error 53 is returned.

SEE ALSO

FindNext
FindPrior
FindLast

FindAtCurRec

SYNTAX

```
Get Jovis (
[1] "FindAtCurRec", ¬
[2] "FileGlobal", ¬
[3] "Key" )
```

WHERE

[1] is the Jovis 'FindAtCurRec' command;
[2] is the global identifier for the file you want to work with;
[3] is the key to locate within the keyset. Partial key strings can also be used.

EXAMPLE

```
on DemoScript
  put Jovis("FindAt",urRec("myDB","Smith") into temp
end DemoScript
```

DESCRIPTION

Finds the first key in the current keyset whose value is equal to the value passed as parameter 2 and which also points to the current record.

SEE ALSO

Currency

FindCur

SYNTAX

```
Put Jovis (
[1] "FindCur", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'FindCur' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  put Jovis("FindCur","myDB") into temp
end DemoScript
```

DESCRIPTION

Returns the current key.

SEE ALSO

Currency

FindCurRec

SYNTAX

```
Put Jovis (
[1] "FindCurRec", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'FindCurRec' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  put Jovis("FindCur",ec("myDB") into temp
end DemoScript
```

DESCRIPTION

Returns the first key in the current keyset that points to the current record.

SEE ALSO

Currency

FindFirst

SYNTAX

```
Put Jovis (
[1] "FindFirst", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'FindFirst' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  put Jovis("FindFirst","myDB") into temp
end DemoScript
```

DESCRIPTION

Returns the first key in the current keyset.

SEE ALSO

GoToKeysetName
GoToKeysetID

FindLast

SYNTAX

```
Put Jovis (
[1] "FindLast", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'FindLast' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  put Jovis("FindLast","myDB") into temp
end DemoScript
```

DESCRIPTION

Returns the last key in the current keyset.

SEE ALSO

GoToKeysetName
GoToKeysetID

FindNext

SYNTAX

```
Put Jovis (
[1] "FindNext", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'FindNext' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  put Jovis("FindNext","myDB") into temp
end DemoScript
```

DESCRIPTION

Returns the next greater key in the current keyset.

SEE ALSO

FindAt
FindFirst
FindPrior

FindPrior

SYNTAX

```
Put Jovis (
[1] "FindPrior", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'FindPrior' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  put Jovis("FindPrior","myDB") into temp
end DemoScript
```

DESCRIPTION

Returns the next smaller key in the current keyset.

SEE ALSO

FindAt
FindFirst
FindNext

FindTop

SYNTAX

```
Put Jovis (
[1] "FindTop", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'FindTop' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  put Jovis("FindTop","myDB") into temp
end DemoScript
```

DESCRIPTION

Returns the first key in the first keyset in the list of keysets.

SEE ALSO

GoToKeysetName
GoToKeysetID

FreeAll

SYNTAX

```
Get Jovis (
[1] "FreeAll", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'FreeAll' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  put Jovis("FreeAll","myDB") into temp
end DemoScript
```

DESCRIPTION

Frees all temporary AND permanent markers.

SEE ALSO

SetMarker
GoToMarker

FreeMarker

SYNTAX

```
Get Jovis (
[1] "FreeMarker", ¬
[2] "FileGlobal", ¬
[3] "MarkerID", ¬
[4] '[MarkerName]' )
```

WHERE

[1] is the Jovis 'FreeMarker' command;
[2] is the global identifier for the file you want to work with;
[3] is the Marker ID. If zero, MarkerName is used.
[4] is the MarkerName is use if Marker ID is zero.

EXAMPLE

```
on DemoScript
  put Jovis("FreeMarker","myDB",21) into temp
  put Jovis("FreeMarker","myDB",0,"OrderMarker") into temp
end DemoScript
```

DESCRIPTION

Frees the marker represented by Marker ID.

COMMENTS

If Marker ID is 0, then the command frees the marker whose name is Marker Name. If the database is open read-only, freeMarker can only free temporary markers.

SEE ALSO

SetMarker
GoToMarker

GetFileName

SYNTAX

```
Put Jovis (
[1] "GetFileName", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'GetFileName' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  put Jovis("GetFileName","thisFile") into temp
end DemoScript
```

DESCRIPTION

Returns the complete path name of the Jovis database file opened with the global identifier given as the second parameter.

GetKeysetID

SYNTAX

```
Put Jovis (
[1] "GetKeysetID", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'GetKeysetID' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  get Jovis("GetKeysetID","myDB")
end DemoScript
```

DESCRIPTION

Returns the keyset ID for the current keyset.

SEE ALSO

GoToKeysetID
GetKeysetName
SetKeysetName

GetKeysetName

SYNTAX

```
Put Jovis (
[1] "GetKeysetName", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'GetKeysetName' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  get Jovis("GetKeysetName","myDB")
end DemoScript
```

DESCRIPTION

Returns the name of the current keyset.

SEE ALSO

GetKeysetID
GoToKeysetName

GetMarkerID

SYNTAX

```
Put Jovis (
[1] "GetMarkerID", ¬
[2] "FileGlobal", ¬
[3] "MarkerName" )
```

WHERE

[1] is the Jovis 'GetMarkerID' command;
[2] is the global identifier for the file you want to work with;
[3] is the Marker name whose ID will be returned.

EXAMPLE

```
on DemoScript
  put Jovis("GetMarkerID","myDB","Zip") into id
end DemoScript
```

DESCRIPTION

Returns the ID of the marker whose name is passed as parameter 2.

SEE ALSO

SetMarker
GoToMarker

GetMarkerName

SYNTAX

```
Put Jovis (
[1] "GetMarkerName", ¬
[2] "FileGlobal", ¬
[3] "MarkerID" )
```

WHERE

[1] is the Jovis 'GetMarkerName' command;
[2] is the global identifier for the file you want to work with;
[3] is the MarkerID whose name will be returned.

EXAMPLE

```
on DemoScript
  put Jovis("GetMarkerName","myData",markerID) into markerName
end DemoScript
```

DESCRIPTION

Returns the name of the marker whose ID is passed as parameter 2.

SEE ALSO

SetMarker
GoToMarker
GetMarkerID

GetVersion

SYNTAX

```
Put Jovis (
[1] "GetVersion", ¬
[2] "FileGlobal", ¬
[3] "True or False" )
```

WHERE

[1] is the Jovis 'GetVersion' command;
[2] is the global identifier for the file you want to work with;
[3] If this parameter is true, getVersion returns the version number of the file format. If it is false, getVersion returns the version number of the Jovis engine.

EXAMPLE

```
on DemoScript
  put Jovis("GetVersion","myDB",true) into temp
  put Jovis("GetVersion","myDB") into temp
end DemoScript
```

DESCRIPTION

Returns a version of either the Jovis file format or the Jovis Engine. The file format version is the version which the engine is compatible with.

COMMENTS

The same information can be gotten using single charachter codes without opening a data file.
For Example:
put Jovis("%") -- returns version of Jovis engine.
put Jovis("*") -- returns version of the file format.

GoToKey

SYNTAX

```
Put Jovis (
[1] "GoToKey", ¬
[2] "FileGlobal", ¬
[3] "Key Position" )
```

WHERE

[1] is the Jovis 'GoToKey' command;
[2] is the global identifier for the file you want to work with;
[3] is a position in the keyset.

EXAMPLE

```
on DemoScript
  put Jovis("GoToKey","myDB",17) into key
end DemoScript
```

DESCRIPTION

Goes to and returns the key in the current keyset, using the position indicated by parameter 2.

COMMENTS

Key positioning does not use key values. Rather, the relative location is based on the total number of keys in the keyset. For example, the 23 would be the 23rd key in the keyset.

GoToKeysetID

SYNTAX

```
Get Jovis (
[1] "GoToKeysetID", ¬
[2] "FileGlobal", ¬
[3] "KeysetID" )
```

WHERE

[1] is the Jovis 'GoToKeySetID' command;
[2] is the global identifier for the file you want to work with;
[3] is the ID of the keyset to make current.

EXAMPLE

```
on DemoScript
  get Jovis("GoToKeysetID","myDB","5")
end DemoScript
```

DESCRIPTION

Make the keyset indicated by the keysetID in parameter 3, the current keyset.

COMMENTS

If there are keys in the keyset, the first key is made the current key.

SEE ALSO

ListKeysetNames
GoToKeysetName

GoToKeysetName

SYNTAX

```
Get Jovis (
[1] "GoToKeySetName", ¬
[2] "FileGlobal", ¬
[3] "KeysetName" )
```

WHERE

[1] is the Jovis 'GoToKeysetName' command;
[2] is the global identifier for the file you want to work with;
[3] is the name of the keyset to make current.

EXAMPLE

```
on DemoScript
  get Jovis("GoToKeysetName","myDB","ItemPicture")
end DemoScript
```

DESCRIPTION

Make the keyset indicated by the keyset name in parameter 3, the current keyset.

COMMENTS

If there are keys in the keyset, the first key is made the current key.

SEE ALSO

ListKeysetNames
GoToKeysetID

GoToMarker

SYNTAX

```
Get Jovis (
[1] "GoToMarker", ¬
[2] "FileGlobal", ¬
[3] "MarkerID", ¬
[4] '[MarkerName]' )
```

WHERE

[1] is the Jovis 'GoToMarker' command;
[2] is the global identifier for the file you want to work with;
[3] is the marker ID to use,
[4] is the marker name to use if marker ID is zero.

EXAMPLE

```
on DemoScript
  put Jovis("GoToMarker","myDB",42) into temp
  put Jovis("GoToMarker","myDB",0,"ZipMark") into temp
end DemoScript
```

DESCRIPTION

Positions you at the key (or keyset) represented by Marker ID.

COMMENTS

If Marker ID is 0, then the command uses Marker Name instead. (See SetMarker.)

GoToTop

SYNTAX

```
Get Jovis (
[1] "GoToTop", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'GoToTop' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  put Jovis("GoToTop","myDB") into temp
end DemoScript
```

DESCRIPTION

Positions you to the first key in the first keyset, without returning a key value.

COMMENTS

No longer used.

SEE ALSO

GoToKeysetID
GoToKeysetName

KeysetStatus

SYNTAX

```
Put Jovis (
[1] "KeysetStatus", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'KeysetStatus' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  put Jovis("KeysetStatus","myDB") into temp
end DemoScript
```

DESCRIPTION

Returns a comma-separated list of information about the current keyset: KeysetName,KeyLength,Duplicate keys allowed,KeyKind,KeysetID,Number of keys in keyset,Number of BTree nodes,Internally created keyset.

COMMENTS

See demo script for a simple way to get information about all keysets.

SEE ALSO

KeyStatus

KeyStatus

SYNTAX

```
Put Jovis (
[1] "KeyStatus", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'KeyStatus' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  put Jovis("KeyStatus","myDB") into temp
end DemoScript
```

DESCRIPTION

Returns a comma-separated list of information about the current key: HasRecHdr, RecType, KeyID, Keysize, HdrLoc,RecLoc,RecSize, RefCon, Key

Note: The 'RefCon' item is for future use.

SEE ALSO

KeysetStatus

ListKeysetNames

SYNTAX

```
Put Jovis (
[1] "ListKeysetNames", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'ListKeysetNames' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  get Jovis("ListKeysetNames","myDB")
end DemoScript
```

DESCRIPTION

Returns a two item list of the names and IDs of all keysets.

SEE ALSO

GoToKeysetName
GoToKeysetID

LinkRec

SYNTAX

```
Get Jovis (
[1] "LinkRec", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'LinkRec' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  get Jovis("LinkRec","myDB")
end DemoScript
```

DESCRIPTION

Links the current key to the already existing current record.

COMMENTS

If you write a key and then write a record, the two are NOT linked until you call this command. On the other hand, if you write a record first, and then write a key, they will be automatically linked. The secret is that the key "grabs" the current record in the process of being written to the keyset. (This is a very efficient way to have things work.) The LinkRec is most often used when you want to link more than one key, say from a different keyset, to the same record. This concept is refered to as HyperText; having more than one reference point (i.e. key) returning the same record.

SEE ALSO

WriteText
WriteKey

MarkerStatus

SYNTAX

```
Put Jovis (
[1] "MarkerStatus", ¬
[2] "FileGlobal", ¬
[3] '[INTPK]' )
```

WHERE

[1] is the Jovis 'MarkerStatus' command;
[2] is the global identifier for the file you want to work with;
[3] is a string of up to 5 characters indicating the information to return:
I - Marker ID
N - Marker Name
T - True if marker is temporary, false if it is not
P - Key which points to this marker
K - Key which this marker points to

EXAMPLE

```
on DemoScript
  put Jovis("MarkerStatus","myDB","INP") into temp
  put Jovis("MarkerStatus","myDB") into temp
end DemoScript
```

DESCRIPTION

Returns information about each marker in the current keyset.

COMMENTS

When the information is returned, items are separated by commas, and sets of marker information are separated by returns. You do not need to include all five characters in parameter 2. Only include the characters for the information you want returned. If parameter 2 is not included, the default is “INTPK”.

SEE ALSO

SetMarker
GoToMarker

PictToClip

SYNTAX

```
Get Jovis (
[1] "PictToClip", ¬
[2] "PictGlobal" )
```

WHERE

[1] is the Jovis 'PictToClip' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  put Jovis("PictToClip","pictVar") into temp
end DemoScript
```

DESCRIPTION

Puts the picture pointed to by the PictGlobal onto the clipboard.

SEE ALSO

ReadPict
WritePict
DisplayPict
ClipToPict

PictToResID

SYNTAX

```
Get Jovis (
[1] "PictToResID", ¬
[2] "PictGlobal", ¬
[3] '[ResourceName]', ¬
[4] "ResourceID", ¬
[5] '[OverwriteFlag]', ¬
[6] '[FilePath]' )
```

WHERE

[1] is the Jovis 'PictToResID' command;
[2] is the global identifier for the Pict you want to work with;
[3] is an optional name for the 'Pict';
[4] is a required ID for the new reousrce;
[5] if the overwrite flag is "True" and a 'Pict' resource with the given ID exists it is replaced. If the overwrite flag is empty or not "True", and the resource by the ID exists, an error is returned;
[6] is the entire path name to the file to add the resource. If this parameter is empty or missing, the standard get file dialog is displayed.

EXAMPLE

```
on DemoScript
  global thePict
  put "Jovis" into thePict
  get Jovis("PictToResID","thePict","Daisy","663","True")
  get Jovis("PictToResID","thePict","","775","False","HD255:Picts:garden
        flowers")
end DemoScript
```

DESCRIPTION

Creates a new PICT resource from a Jovis global. If a PICT resource of the given ResourceID exists, it is removed if the overwrite FLAG is true. The default for overwrite is false.

COMMENTS

The file to insert the 'Pict' must not be currently open during this operation.

SEE ALSO

PictToResName

PictToResName

SYNTAX

```
Get Jovis (
[1] "PictToResName", ¬
[2] "PictGlobal",
[4] "ResourceID", ¬
[5] '[OverwriteFlag]', ¬
[6] '[FilePath]' )
```

WHERE

[1] is the Jovis 'PictToResName' command;
[2] is the global identifier for the Pict you want to work with;
[3] is an optional name for the 'Pict';
[4] if the overwrite flag is "True" and a 'Pict' resource with the given name exists it is replaced. If the overwrite flag is empty or not "True", and the resource by the name exists, an error is returned;
[5] is the entire path name to the file to add the resource. If this parameter is empty or missing, the standard get file dialog is displayed.

EXAMPLE

```
on DemoScript
  global thePict
  put "Jovis" into thePict
  get Jovis("PictToResName","thePict","Daisy","True")
  get Jovis("PictToResName","thePict","Daisy","False","HD255:Picts:garden
        flowers")
end DemoScript
```

DESCRIPTION

Creates a new PICT resource from a Jovis global. If a PICT resource of the given Resource name exists, it is removed if the overwrite flag is "True". The default for overwrite is "False".

SEE ALSO

PictToResID

PlaySound

SYNTAX

```
Get Jovis (
[1] "PlaySound", ¬
[2] "SndGlobal" )
```

WHERE

[1] is the Jovis 'PlaySound' command;
[2] is the global identifier for the 'snd ' you want to work with;

EXAMPLE

```
on DemoScript
  get Jovis("playSound","soundVar")
end DemoScript
```

DESCRIPTION

Plays the sound pointed to by the Soundvariable.

SEE ALSO

ReadSound
WriteSound
SoundToClip
ResNameToSound

ReadBlob

SYNTAX

```
Get Jovis (
[1] "ReadBlob", ¬
[2] "FileGlobal", ¬
[3] "BlobGlobal", ¬
[4] '[CurRec Flag]', ¬
[5] '[Lock Flag]' )
```

WHERE

[1] is the Jovis 'ReadBlob' command;
[2] is the global identifier for the file you want to work with;
[3] is the global identifier for the 'Blob' you want to work with;
[4] If Current Record is true, ReadBlob does not make the record it reads the current record. If it is false or missing, the record read becomes the current record;
[5] the Lock, has no effect in the single-user version. See comments below.

EXAMPLE

```
on DemoScript
  global BlobGlo
  if BlobGlo is empty then
    put "Jovis" into BlobGlo
    get Jovis("ReadBlob","myDB","BlobGlo")
  end if
end DemoScript
```

DESCRIPTION

Loads the Blob record pointed to by the current key into a Jovis global variable.

COMMENTS

Variations for multi-user version:
If the Lock parameter is true, then the record is locked when it is read. This means that other users on the network may read the record, but they may NOT update it.

If Lock is false or missing, then the record is not locked when it is read. If it was previously locked by the same user, it becomes unlocked.

SEE ALSO

Initializing a Jovis Global
WriteBlob
BlobToResName
Currency

ReadPict

SYNTAX

```
Get Jovis (
[1] "ReadPict", ¬
[2] "FileGlobal", ¬
[3] "PictGlobal", ¬
[4] '[CurRec Flag]', ¬
[5] '[Lock Flag]' )
```

WHERE

[1] is the Jovis 'ReadPict' command;
[2] is the global identifier for the file you want to work with;
[3] is the global identifier for the 'Pict' you want to work with;
[4] If Current Record is true, ReadPict does not make the record it reads the current record. If it is false or missing, the record read becomes the current record;
[5] the Lock, has no effect in the single-user version. See comments below.

EXAMPLE

```
on DemoScript
  global thePict
  if thePict is empty then
    put"Jovis" into thePict
    get Jovis("readPict","myDB","thePict")
  end if
end DemoScript
```

DESCRIPTION

Loads the Pict record pointed to by the current key into a Jovis global variable.

COMMENTS

Variations for multi-user version:
If the Lock parameter is true, then the record is locked when it is read. This means that other users on the network may read the record, but they may NOT update it. If 'Lock' is false or missing, then the record is not locked when it is read. If it was previously locked by the same user, it becomes unlocked.

SEE ALSO

Initializing a Jovis Global
Currency
WritePict

ReadSound

SYNTAX

```
Get Jovis (
[1] "ReadSound", ¬
[2] "FileGlobal", ¬
[3] "SndGlobal", ¬
[4] '[CurRec Flag]', ¬
[5] '[Lock Flag]' )
```

WHERE

[1] is the Jovis 'ReadSound' command;
[2] is the global identifier for the file you want to work with;
[3] is the global identifier for the 'snd ' you want to work with;
[4] If Current Record is true, ReadSound does not make the record it reads the current record. If it is false or missing, the record read becomes the current record;
[5] the Lock, has no effect in the single-user version. See comments below.

EXAMPLE

```
on DemoScript
  global soundVar
  put"Jovis" into soundVar
  get Jovis("readSound","myDB","SoundVar")
end DemoScript
```

DESCRIPTION

Loads the Sound record pointed to by the current key into a Jovis global variable.

COMMENTS

Variations for multi-user version:

If the Lock parameter is true, then the record is locked when it is read. This means that other users on the network may read the record, but they may NOT update it. If 'Lock' is false or missing, then the record is not locked when it is read. If it was previously locked by the same user, it becomes unlocked.

SEE ALSO

Initializing a Jovis Global
Currency
WriteSound

ReadText

SYNTAX

```
Put Jovis (
[1] "ReadText", ¬
[2] "FileGlobal", ¬
[3] [CurRec], ¬
[4] [RelationName] ¬
[5] '[Lock]' )
```

WHERE

[1] is the Jovis 'ReadText' command;
[2] is the global identifier for the file you want to work with;
[3] If `UseCurRec` is true, ReadText does not make the record it reads the current record. If it is false or missing, the record read becomes the current record;
[5] If the current keyset is relational, providing the relation name cuts down on search time;
[4] the Lock, has no effect in the single-user version. See comments below.

EXAMPLE

```
on DemoScript
  put Jovis("ReadText","myDB",true,true) into rec
  get Jovis("ReadText","myDB")
end DemoScript
```

DESCRIPTION

Returns the text record pointed to by the current key. If the current keyset is relatoinal, the relational record is returned. It can then be used by any of the other relational cmmands, such as GetRecordField, UpdateRecord, etc.

COMMENTS

If the Lock parameter is true, then the record is locked when it is read. This means that other users on the network may read the record, but they may NOT update it. If 'Lock' is false or missing, then the record is not locked when it is read. If it was previously locked by the same user, it becomes unlocked.

SEE ALSO

Initializing a Jovis Global
Currency
WriteSound

ReleaseLocks

SYNTAX

```
Get Jovis (
[1] "ReleaseLocks", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'ReleaseLocks' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  get Jovis("releaseLocks","myDB")
end DemoScript
```

DESCRIPTION

Releases all locked records for this client, for the file indicated.

COMMENTS

This command is non-functional in the single-user version of Jovis .

The mechanism for unlocking a single record is to read that record with the lock parameter set to false.

SEE ALSO

Transaction Commands

ResetKeyLength

SYNTAX

```
Get Jovis (
[1] "ResetKeyLength", ¬
[2] "FileGlobal", ¬
[3] "New Length" )
```

WHERE

[1] is the Jovis 'ResetKeyLength' command;
[2] is the global identifier for the file you want to work with;
[3] is the new length for the keys in the keyset.

EXAMPLE

```
on DemoScript
  get Jovis("ResetKeyLength","myData",12)
  get Jovis("ResetKeyLength","myData",keysize)
end DemoScript
```

DESCRIPTION

Resets the maximum length of the keys in the current keyset. (Text keys only.) Also, if duplicates are NOT allowed in the keyset, this command cannot be used.

COMMENTS

Currently taken off-line. Will be available in future versions of Jovis greater than v1.0.

If you are changing a lot of keys, this command may take a long time. You cannot interrupt it without corrupting your database; it must run to completion. Always work with copy of your data file!

ResIDToPict

SYNTAX

```
Get Jovis (
[1] "ResIDToPict", ¬
[2] "PictGlobal", ¬
[3] '[ResSelector Coors]', ¬
[4] '[File PathName]', ¬
[5] '[Resource Name]' )
```

WHERE

[1] is the Jovis 'ResIDToPict' command;
[2] is the global identifier for the 'Pict' you want to work with;
[3] is the optional coordinates for the resource picker dialog. The default is "100,100";
[4] is the entire path name to the file to read the 'Pict' resource from. If this parameter is empty or missing, the standard get file dialog is displayed;
[5] is the name of the resource to retrieve.

EXAMPLE

```
on DemoScript
  global thePict
  put"Jovis" into thePict
  get Jovis("ResIDToPict","thePict","663")
  get Jovis("ResIDToPict","thePict","","200,150","HD255:Picts:garden flow-
        ers")
end DemoScript
```

DESCRIPTION

Takes a PICT resource and converts it to a Jovis global variable.

SEE ALSO

Initializing a Jovis Global
ResNameToPict
DisplayPict
WritePict

ResIDToSound

SYNTAX

```
Get Jovis (
[1] "ResIDToSound", ¬
[2] "PictGlobal", ¬
[3] '[ResSelector Coors]', ¬
[4] '[File PathName]', ¬
[5] '[Resource Name]' )
```

WHERE

[1] is the Jovis 'ResIDToSound' command;
[2] is the global identifier for the 'snd ' you want to work with;
[3] is the optional coordinates for the resource picker dialog. The default is "100,100";
[4] is the entire path name to the file to read the 'snd ' resource from. If this parameter is empty or missing, the standard get file dialog is displayed;
[5] is the name of the resource to retrieve. If this parameter is empty a resouce picker is displayed.

EXAMPLE

```
on DemoScript
  global theSnd
  put"Jovis" into theSnd
  get Jovis("ResIDToSound","theSnd","243")
  get Jovis("ResIDToSound","theSnd","","100,100","HD255:Sounds:Voices")
end DemoScript
```

DESCRIPTION

Takes a 'snd ' resource and converts it to a Jovis global variable.

SEE ALSO

Initializing a Jovis Global
ResNameToSound

ResNameToPict

SYNTAX

```
Get Jovis (
[1] "ResNameToPict", ¬
[2] "PictGlobal", ¬
[3] '[ResSelector Coors]', ¬
[4] '[File PathName]', ¬
[5] '[Resource Name]' )
```

WHERE

[1] is the Jovis 'ResNAmeToPict' command;
[2] is the global identifier for the 'Pict' you want to work with;
[3] is the optional coordinates for the resource picker dialog. The default is "100,100";
[4] is the entire path name to the file to read the 'Pict' resource from. If this parameter is empty or missing, the standard get file dialog is displayed;
[5] is the name of the resource to retrieve. If this parameter is empty a resouce picker is displayed.

EXAMPLE

```
on DemoScript
  global thePict
  put"Jovis" into thePict
  get Jovis("ResNameToPict","thePict","flower")
  get Jovis("ResNameToPict","thePict","","150,200","HD255:Picts:garden
        flowers")
end DemoScript
```

DESCRIPTION

Takes a PICT resource and converts it to a Jovis global variable.

SEE ALSO

Initializing a Jovis Global
ResIDToPict
DisplayPict
WritePict

ResNameToSound

SYNTAX

```
Get Jovis (
[1] "ResNameToSound", ¬
[2] "SndGlobal", ¬
[3] '[ResSelector Coors]', ¬
[4] '[File PathName]', ¬
[5] '[Resource Name]' )
```

WHERE

[1] is the Jovis 'ResNameToPict' command;
[2] is the global identifier for the 'Pict' you want to work with;
[3] is the optional coordinates for the resource picker dialog. The default is "100,100";
[4] is the entire path name to the file to read the 'Pict' resource from. If this parameter is empty or missing, the standard get file dialog is displayed;
[5] is the name of the resource to retrieve. If this parameter is empty a resouce picker dialog is displayed.

EXAMPLE

```
on DemoScript
  global theSnd
  put"Jovis" into theSnd
  get Jovis("ResNameToSound","theSnd","music")
  get Jovis("ResNameToSound","theSnd","","100,100","HD255:Sounds:Voices")
end DemoScript
```

DESCRIPTION

Takes a PICT resource and converts it to a Jovis global variable.

SEE ALSO

Initializing a Jovis Global
ResIDToPict
DisplayPict
WritePict

ResToBlob

SYNTAX

```
Get Jovis (
[1] "ResToBlob", ¬
[2] "FileGlobal", ¬
[3] "ResType", ¬
[4] '[ResID]', ¬
[5] '[Coors]', ¬
[6] '[File PathName]' )
```

WHERE

[1] is the Jovis 'ResToBlob' command;
[2] is the global identifier for the 'Pict' you want to work with;
[3] is the resource type you are retrieving;
[4] is the optional resource ID. If this parameter is zero or empty, the resource picker is displayed;
[5] is the optional coordinates for the resource picker dialog. The default is "100,100";
[6] is the entire path name to the file to read the 'Blob' resource from. If this parameter is empty or missing, the standard get file dialog is displayed;

EXAMPLE

```
on DemoScript
  get Jovis("ResToBlob","myDB","ICON")
end DemoScript
```

DESCRIPTION

Takes any resource type and converts it to a Jovis global variable.

SEE ALSO

Initializing a Jovis Global
WriteBlob
RewriteBlob
BlobToResID
BlobToResName

RewriteBlob

SYNTAX

```
Get Jovis (
[1] "RewriteBlob", ¬
[2] "FileGlobal", ¬
[3] "BlobGlobal" )
```

WHERE

[1] is the Jovis 'RewriteBlob' command;
[2] is the global identifier for the file you want to work with;
[3] is the name of the global containing the 'Blob'.

EXAMPLE

```
on DemoScript
  global BlobGlobal
  get Jovis("RewriteBlob","myDB","BlobGlobal")
end DemoScript
```

DESCRIPTION

Replaces the 'Blob' record pointed to by the current key, with the one pointed to by BlobGlobal.

COMMENTS

Variations for multi-user version:

In order to be rewritten, the record must have been locked when it was read. Note that 'RewriteBlob' unlocks the record.

SEE ALSO

ReadBlob
ClearBlob

RewriteKey

SYNTAX

```
Get Jovis (
[1] "RewriteKey", ¬
[2] "FileGlobal", ¬
[3] "Key" )
```

WHERE

[1] is the Jovis 'RewriteKey' command;
[2] is the global identifier for the file you want to work with;
[3] is the new key to replace the current key with.

EXAMPLE

```
on DemoScript
  put Jovis("rewriteKey","myDB", "New Key Value") into returnedValue
end DemoScript
```

DESCRIPTION

Replaces the current key with the new key.

COMMENTS

If the current key is linked to a record, the new key value will also be linked to it.

SEE ALSO

WriteKey
DelKey

RewritePict

SYNTAX

```
Get Jovis (
[1] "RewritePict", ¬
[2] "FileGlobal", ¬
[3] "PictGlobal" )
```

WHERE

[1] is the Jovis 'RewritePict' command;
[2] is the global identifier for the file you want to work with;
[3] is the name of the 'PictGlobal' to replace the current 'Pict' record with.

EXAMPLE

```
on DemoScript
  get Jovis("rewritePict","myDB","pictVar")
end DemoScript
```

DESCRIPTION

Replaces the Pict record pointed to by the current key, with the one pointed to by PictGlobal.

COMMENTS

Variations for multi-user version:

In order to be rewritten, the Pictrecord must have been locked when it was read. Note that 'RewritePict' unlocks the record.

SEE ALSO

ReadPict
WritePict
ClearPict

RewriteSound

SYNTAX

```
Get Jovis (
[1] "RewriteSound", ¬
[2] "FileGlobal", ¬
[3] "SoundGlobal" )
```

WHERE

[1] is the Jovis 'RewriteSound' command;
[2] is the global identifier for the file you want to work with;
[3] is the name of the 'SoundGlobal' to replace the current 'snd ' record with.

EXAMPLE

```
on DemoScript
  get Jovis("rewriteSound","myDB","theSound")
end DemoScript
```

DESCRIPTION

Replaces the sound record pointed to by the current key, with the one pointed to by the SoundGlobal.

COMMENTS

Variations for multi-user version:

In order to be rewritten, the record must have been locked when it was read. RewriteSound unlocks the record.

RewriteText

SYNTAX

```
Get Jovis (
[1] "RewriteText", ¬
[2] "FileGlobal", ¬
[3] "New Text" )
```

WHERE

[1] is the Jovis 'RewriteText' command;
[2] is the global identifier for the file you want to work with;
[3] is the new text string to replace the current text record with.

EXAMPLE

```
on DemoScript
  get Jovis("rewriteText","myDB",cd fld "comments")
end DemoScript
```

DESCRIPTION

Replaces the text record associated with the current key, with the given new text string.

COMMENTS

Variations for multi-user version:

In order to be rewritten, the record must have been locked when it was read. RewriteText unlocks the record.

SEE ALSO

WriteText
WriteKey
RewriteText

SetCurRec

SYNTAX

```
Get Jovis (
[1] "SetCurRec", ¬
[2] "FileGlobal" )
```

WHERE

[1] is the Jovis 'SetCurRec' command;
[2] is the global identifier for the file you want to work with;

EXAMPLE

```
on DemoScript
  get Jovis("setCurRec","myDB")
end DemoScript
```

DESCRIPTION

Makes the record pointed to by the current key, the current record. This allows you to set the current record without having to read it. This command works for Text, Pict, Blob, and Sound records.

SetKeysetName

SYNTAX

```
Get Jovis (
[1] "SetKeysetName", ¬
[2] "FileGlobal", ¬
[3] "KeysetName" )
```

WHERE

[1] is the Jovis 'SetKeysetName' command;
[2] is the global identifier for the file you want to work with;
[3] is the new name for the current keyset.

EXAMPLE

```
on DemoScript
  get Jovis("SetKeysetName","myDB","PhoneNum2")
end DemoScript
```

DESCRIPTION

Renames the current keyset.

SEE ALSO

ListKeysetNames
KeysetStatus

SetMarker

SYNTAX

```
Put Jovis (
[1] "SetMarker", ¬
[2] "FileGlobal", ¬
[3] "Keyset Only Flag", ¬
[4] '[Marker Name]', ¬
[5] '[Temporary Flag]' )
```

WHERE

[1] is the Jovis 'SetMarker' command;
[2] is the global identifier for the file you want to work with;
[3] If Keyset Only is true, only the current keyset is marked. If it is false, then the current key is marked.
[4] If a marker name is given, the marker can later be referred to by that name. The marker name must not exceed 31 characters in length. NOTE: Permanent markers must be named. Temporary markers may be named, but it is not required.
[5] If Temporary is set to true, the marker is temporary and will not be saved when the file is closed. If it is false, or missing, the marker is permanent and is saved to the database.

EXAMPLE

```
on DemoScript
  put Jovis("setMarker","myDB",False) into markerID
  put Jovis("setMarker","myDB",True,"",True) into markerID
  put Jovis("setMarker","myDB",False,"Last Key Read",  True) into markerID
end DemoScript
```

DESCRIPTION

Marks the current position in the database and returns an ID number which represents that marker.

COMMENTS

NOTE: Permanent markers must be named. Temporary markers may be named, but it is not required.

NOTE: Permanent markers can only be created or freed if the database is open read/write.

For the multi-user version:

Permanent markers are available to any user of the database. Temporary markers are available only to the user that created them.

SEE ALSO

GoToMarker
FreeMarker

SoundToClip

SYNTAX

```
Get Jovis (
[1] "SoundToClip", ¬
[2] "SndGlobal" )
```

WHERE

[1] is the Jovis 'SoundToClip' command;
[2] is the name of the 'snd ' global you want to copy to the clipboard;

EXAMPLE

```
on DemoScript
  get Jovis("SoundToClip","SndGlobal")
end DemoScript
```

DESCRIPTION

Puts the sound pointed to by the SndGlobal onto the clipboard.

COMMENTS

Because sounds can be quite large, you must be sure that you have enough memory allocated to the shell application.

SEE ALSO

ClearSound

SoundToResID

SYNTAX

```
Get Jovis (
[1] "SoundToResID", ¬
[2] "PictGlobal", ¬
[3] '[ResourceName]', ¬
[4] "ResourceID", ¬
[5] '[OverwriteFlag]', ¬
[6] '[FilePath]' )
```

WHERE

[1] is the Jovis 'SoundToResID' command;
[2] is the global identifier for the 'snd ' you want to work with;
[3] is an optional name for the 'snd ';
[4] is a required ID for the new reousrce;
[5] if the overwrite flag is "True" and a 'snd ' resource with the given ID exists it is replaced. If the overwrite flag is empty or not "True", and the resource by the ID exists, an error is returned;
[6] is the entire path name to the file to add the resource. If this parameter is empty or missing, the standard get file dialog is displayed.

EXAMPLE

```
on DemoScript
  get Jovis("SoundToResID","theSnd","","39567","True")
  get Jovis("SoundToResID", "theSnd", "Voice", "1234", "True",
         "HD255:Sounds:Voice1")
end DemoScript
```

DESCRIPTION

Creates a new 'snd ' resource from a Jovis global. If a 'snd ' resource of the given ResourceID exists, it is removed if the overwrite FLAG is true. The default for overwrite is false.

COMMENTS

The file to insert the 'snd ' must not be open during this operation.

SEE ALSO

ClearSound

SoundToResName

SYNTAX

```
Get Jovis (
[1] "SoundToResName", ¬
[2] "PictGlobal",
[4] "ResourceID", ¬
[5] '[OverwriteFlag]', ¬
[6] '[FilePath]' )
```

WHERE

[1] is the Jovis 'SoundToResName' command;
[2] is the global identifier for the 'snd ' you want to work with;
[3] is an optional name for the 'snd ';
[4] if the overwrite flag is "True" and a 'snd ' resource with the given name exists it is replaced. If the overwrite flag is empty or not "True", and the resource by the name exists, an error is returned;
[5] is the entire path name to the file to add the resource. If this parameter is empty or missing, the standard get file dialog is displayed.

EXAMPLE

```
on DemoScript
  get Jovis("SoundToResName","theSnd","Flipper","True")
  get Jovis("SoundToResName", "theSnd", "MaleVoice", "True",
         "HD255:Sounds:Voice1")
end DemoScript
```

DESCRIPTION

Creates a new 'snd ' resource from a Jovis global. If a 'snd ' resource of the given Resource name exists, it is removed if the overwrite flag is "True". The default for overwrite is "False".

SEE ALSO

ClearSound

WriteBlob

SYNTAX

```
Get Jovis (
[1] "WriteBlob", ¬
[2] "FileGlobal", ¬
[3] "BlobGlobal" )
```

WHERE

[1] is the Jovis 'WriteBlob' command;
[2] is the global identifier for the file you want to work with;
[3] is the name of the 'BlobGlobal' to write to the database.

EXAMPLE

```
on DemoScript
  get Jovis("WriteBlob","myDB","BlobGlobal")
end DemoScript
```

DESCRIPTION

Writes the Blob record to the database. The new Blob record becomes the current record.

SEE ALSO

ClearBlob
Readblob

WriteKey

SYNTAX

```
Get Jovis (
[1] "WriteKey", ¬
[2] "FileGlobal", ¬
[3] "Key", ¬
[4] '['UseCurRec']' )
```

WHERE

[1] is the Jovis 'WriteKey' command;
[2] is the global identifier for the file you want to work with;
[3] is the key to insert into the keyset;
[4] is a string literal indicating whether or not to link the current record to the new key. If this parameter is the string "UseCurRec", the current record will be linked with the newly created key. Otherwise, no record will be linked at this time.

EXAMPLE

```
on DemoScript
  get Jovis("WriteKey","myDB","A_Key_Value", "UseCurRec")
end DemoScript
```

DESCRIPTION

Writes the key to the current keyset and makes it the current key.

SEE ALSO

WriteText
DefineKeyset
LinkRec

WritePict

SYNTAX

```
Get Jovis (
[1] "WritePict", ¬
[2] "FileGlobal", ¬
[3] "PictGlobal" )
```

WHERE

[1] is the Jovis 'WritePict' command;
[2] is the global identifier for the file you want to work with;
[3] is the name of the 'PictGlobal' to write to the database.

EXAMPLE

```
on DemoScript
  get Jovis("writePict","myDB","thePict")
end DemoScript
```

DESCRIPTION

Writes the Pict record to the database. The new 'Pict' record becomes the current record.

SEE ALSO

WriteKey
LinkRec

WriteSound

SYNTAX

```
Get Jovis (
[1] "WriteSound", ¬
[2] "FileGlobal", ¬
[3] "SndGlobal" )
```

WHERE

[1] is the Jovis 'WriteSound' command;
[2] is the global identifier for the file you want to work with;
[3] is the name of the 'SndGlobal' to write to the database.

EXAMPLE

```
on DemoScript
  get Jovis("writeSound","myDB","theSound")
end DemoScript
```

DESCRIPTION

Writes the sound record to the database.

WriteText

SYNTAX

```
Get Jovis (
[1] "WriteText", ¬
[2] "FileGlobal", ¬
[3] "Text Record" )
```

WHERE

[1] is the Jovis 'WriteText' command;
[2] is the global identifier for the file you want to work with;
[3] is the text you want to write to the database.

EXAMPLE

```
on DemoScript
  get Jovis("WriteText","myDB",cd fld "Stuff")
end DemoScript
```

DESCRIPTION

Writes the text record to the database. The new text record becomes the current record.

SEE ALSO

WriteKey
DefineKeyset
ReWriteText