

# Jovis Utility Commands

## Syntax Reference



## ChunkOffset

### SYNTAX

```
Put Jovis (  
[1] "ChunkOffset", ↵  
[2] "Target", ↵  
[3] "SearchFor", ↵  
[4] '[Occurrence]' )
```

### WHERE

[1] is the Jovis ChunkOffset command;  
[2] is the global identifier for the file you want to work with;  
[3] is the name of the relation you want to work with within the designated file;  
[4] Occurrence is which occurrence of the string to find.  
If the parameter is empty or missing, the first occurrence is returned.

### EXAMPLE

```
on DemoScript  
  put line 2 of Jovis("chunkOffset",fld "Target","Apple") into theOffset  
  put item 1 of line 4 of Jovis("chunkOffset",fld "To Search",fld  
    "Comments") into bin  
end DemoScript
```

### DESCRIPTION

Analogous to the HyperTalk Offset command.

### COMMENTS

The results returned are as follows:

character offset in line 1  
word offset in line 2  
item offset in line 3  
line offset in item 1 of line 4  
character within the line in item 2 of line 4  
word within the line in item 3 of line 4  
item offset within the line in item 4 of line 4

### SEE ALSO

test 1

## DBSaveACopy

### SYNTAX

```
Get Jovis (
[1] "DBSaveACopy", ↵
[2] '[Destination PathName]', ↵
[3] '[Overwrite Flag]', ↵
[4] '[Prompt]', ↵
[5] '[Destination FileName]', ↵
[6] '[Dialog Coordinates]' )
```

### WHERE

- [1] is the Jovis DBSaveACopy command;
- [2] is the fileGlobal to the database that you want to copy. NOTE: The fileGlobal identifies the file you wish to copy; you do not need to name it in another parameter;
- [2] entire pathname of the db destination file. If the parameter is empty or missing, standard put dialog box is displayed to prompt for the new file name;
- [3] is an overwrite flag. If the overwrite flag is false or missing, the destination file will not be overwritten if it already exists. If the overwrite flag is "true" and the destination file exists, it will be overwritten. If the destination parameter is missing and the standardput dialog box is used to get the destination file name, the overwrite flag is ignored;
- [4] is an optional prompt for the standardputfile dialog box. If it is empty or missing, the prompt "Enter new file name:" is used;
- [5] is an optional default file name for the destination file. This is the name that will appear in the PutFile dialog box. If the parameter is missing or empty, then the string "Copy of " is added to the front of the database name to form the destination file name. If the length of the destination file name exceeds 31 characters, it is truncated.
- [6] Dialog Box Coordinates; is a parameter that allows you to specify where the dialog box is displayed on your monitor. The parameter holds the coordinates of the top left corner of the dialog box. If the parameter is empty or missing, then the default coordinates are 60,60.

### EXAMPLE

```
on DemoScript
  get Jovis("DBSaveACopy", "myDB")
  get Jovis("DBSaveACopy", "myDB", "HD:Backup DB", True)
  get Jovis("DBSaveACopy", "myDB", empty, empty, "Select name for backup
    file", "My DB Backup", "125,210")
end DemoScript
```

## DESCRIPTION

Saves a copy of a currently open Jovis database file.

## COMMENTS

This command works with both architectural and Relational databases.

DBSaveACopy cannot be used with the multi-user version to make a copy of a database file at the Server.

## FormatVar

### SYNTAX

```
Put Jovis (  
[1] "FormatVar", ↵  
[2] "Input String", ↵  
[3] '[Field Delim]', ↵  
[4] '[Rec Delim]', ↵  
[5] '[Prestring]', ↵  
[6] '[Poststring]', ↵  
[7] "f1", ↵  
[8] "f2", "... " ↵  
[9] "f12" )
```

### WHERE

- [1] is the Jovis FormatVar command;
- [2] The input string to be formatted;
- [3] Field delimiter; is the ASCII character used to separate fields in the incoming string. The default value is a comma (ASCII 44) character;
- [4] Record delimiter; is the ASCII character used to separate records in the incoming string. The default value is the return (ASCII 13) character;
- [5] Prestring is a string to precede every output record. The default is an empty string;
- [6] Poststring is a string to follow every output record. The default is an empty string;
- [7] The f1, f2, f3, ..., f12 are the field formatting parameters. You can have a minimum of one, and a maximum of twelve. Each parameter contains three items:
  - the field size (no default),
  - the justification (R,L,C - default is L), and
  - the separation string (default is empty string; i.e., no separation after the field).The separation string is the string that follows the field being displayed. It can be any combination of characters that you wish.

### EXAMPLE

```
on DemoScript  
  put formatVar(theData,"","","","","25,L","10,C","10,R") into theVar  
  put Jovis("selectionToVar","myDB","Customers",tab,return,"Title,Year")  
    bin  
  put formatVar(bin,tab,return,empty,empty,"30,L","5,R") into fld "exam-  
    ple"  
end DemoScript
```

## DESCRIPTION

This command reformats the data in a container, to allow you to display it as a table, or however you want it to look. It works on any container, and is not restricted to Jovis data.

## COMMENTS

The data being formatted **MUST** be displayed in a font with mono-spaced characters, such as Monaco. It will not return good results for proportional fonts.

If the data in a field is longer than the field size item in the parameter, the data is truncated at the number of characters given.

If the data is shorter than the field size item, the data is padded with spaces, to the size specified.

## GetField

### SYNTAX

```
Put Jovis (  
  [1] "GetField", ↵  
  [2] "Container", ↵  
  [3] "Field Numer", ↵  
  [4] '[Delimiter]' )
```

### WHERE

- [1] is the Jovis GetField command;
- [2] is the input data set;
- [3] is the field number;
- [4] is the optional field delimiter. The Default is a comma.

### EXAMPLE

```
on DemoScript  
  put Jovis("getField",someString,4) into aField  
  put Jovis("getField",aString,2,return) into aLine  
end DemoScript
```

### DESCRIPTION

Extracts a chunk of a container, using parameter 4 as a delimiter.

### SEE ALSO

PutField



## GetGlobal

### SYNTAX

```
Put Jovis (
  [1] "GetGlobal", ↵
  [2] "Global Name" )
```

### WHERE

[1] is the Jovis GetGlobal command;  
[2] is the global name to use.

### EXAMPLE

```
on DemoScript
  put Jovis("getGlobal","someGlobalVar") into value
end DemoScript
```

### DESCRIPTION

Returns the value of the global variable whose name is passed as a parameter.

### SEE ALSO

SetGlobal

## LowerCase

### SYNTAX

```
Put Jovis (
  [1] "LowerCase", ↵
  [2] "String" )
```

### WHERE

[1] is the Jovis LowerCase command;  
[2] is the text to change to lower case.

### EXAMPLE

```
on DemoScript
  put Jovis("lowerCase",aString) into aString
end DemoScript
```

### DESCRIPTION

Returns the string with all alphabetic characters converted to lowercase.

### SEE ALSO

UpperCase

## Pad

### SYNTAX

```
Put Jovis (  
[1] "Pad", ↵  
[2] "String", ↵  
[3] "Size", ↵  
[4] "Justification", ↵  
[5] "Pad Character" )
```

### WHERE

- [1] is the Jovis Pad command;
- [2] is the input text;
- [3] is the total length (i.e. size) you want returned;
- [4] Justification is L (left), R (right), or C (center);
- [5] is the character to use for padding the string.

### EXAMPLE

```
on DemoScript  
  put Jovis("Pad","10",6,"R","0") into Key  
end DemoScript
```

### DESCRIPTION

Returns String padded to the number of characters given by Size, with the characters indicated.

### COMMENTS

Jovis “naturally” uses null characters in its keyset as padding to fill out keys that are shorter than the length set by the keyset. This command is handy for padding keys with different character.

## PutField

### SYNTAX

```
Put Jovis (  
  [1] "PutField", ↵  
  [2] "container", ↵  
  [3] "Field Number", ↵  
  [4] '[Delimiter]', ↵  
  [5] '[Value]' )
```

### WHERE

- [1] is the Jovis PutField command;
- [2] is the input container to use;
- [3] is which field to update;
- [4] is the delimiter to used by the container;
- [5] is the new value to update the field with.

### EXAMPLE

```
on DemoScript  
  put Jovis("putField",aRecord,4,";", "new value")into aRecord  
end DemoScript
```

### DESCRIPTION

Replaces a chunk of a container, using parameter 4 as a delimiter, with Value.

### SEE ALSO

GetField

## SetGlobal

### SYNTAX

```
Get Jovis (  
  [1] "SetGlobal", ↵  
  [2] "GlobalName", ↵  
  [3] "Value" )
```

### WHERE

- [1] is the Jovis SetGlobal command;
- [2] is the name of the global to update;
- [3] is the value to update the global with.

### EXAMPLE

```
on DemoScript  
  put Jovis("setGlobal","aGlobalVar","a Value") into temp  
end DemoScript
```

### DESCRIPTION

Sets the global variable named in parameter 1 to the value passed as parameter 2.

### COMMENTS

This command dynamically creates and loads a shell application's global. There may be times when you don't know the name of the global you need to work with until your script starts to execute.

### SEE ALSO

GetGlobal

## StripIt

### SYNTAX

```
Put Jovis (
  [1] "StripIt", ↵
  [2] "Input String" )
```

### WHERE

- [1] is the Jovis StripIt command;
- [2] is the input string to use for stripping non-numeric characters.

### EXAMPLE

```
on DemoScript
  put Jovis("stripIt",aString) into aString
end DemoScript
```

### DESCRIPTION

Returns a String with all non-numeric characters removed.

### COMMENTS

Useful for removing formatting characters, such as dollar signs.

## Substitute

### SYNTAX

```
Put Jovis (  
  [1] "Substitute", ↵  
  [2] "Input Container", ↵  
  [3] "CharToFind", ↵  
  [4] "CharToInstall" )
```

### WHERE

- [1] is the Jovis Substitute command;
- [2] is the input string;
- [3] is the character to find;
- [4] is the character to replace.

### EXAMPLE

```
on DemoScript  
  put Jovis("substitute",aString,comma,";") into aString  
end DemoScript
```

### DESCRIPTION

Returns a string that replaces all occurrences of a given character with another.

## Trim

### SYNTAX

```
Put Jovis (  
  [1] "Trim", ↵  
  [2] "Input String" )
```

### WHERE

[1] is the Jovis Trim command;  
[2] is the input string to trim.

### EXAMPLE

```
on DemoScript  
  put Jovis("Trim",aString) into aKey  
end DemoScript
```

### DESCRIPTION

Returns String with any leading spaces and zeroes, and any trailing spaces, removed.



## UpperCase

### SYNTAX

```
Put Jovis (  
  [1] "UpperCase", ↵  
  [2] "Input String" )
```

### WHERE

[1] is the Jovis UpperCase command;  
[2] is the input string to have uppercased.

### EXAMPLE

```
on DemoScript  
  put Jovis("uppercase",lowerString) into upperString  
end DemoScript
```

### DESCRIPTION

Returns String with all of the alphabetic characters converted to uppercase.

### SEE ALSO

LowerCase

## ZapFile

### SYNTAX

```
Get Jovis (
[1] "ZapFile", ↵
[2] '[Full PathName]' )
```

### WHERE

- [1] is the Jovis ZapFile command;
- [2] is the entire pathname to a file to delete from the hard drive. If this parameter is empty, or missing the get standard file dialog is displayed.

### EXAMPLE

### DESCRIPTION

Delete any file on a local hard drive.

### COMMENTS

You can use this command to delete any file, not just a database file. Needless to say, some caution should be used with this command. There is NO undo option!

