Built-in BobCAD-CAM Post Processor Lua Functions

Introduction

This topic contains the reference for built-in BobCAD-CAM Post Processor Lua Functions. These functions can be used through-out the post processor to add more customized modifications to the Post Variables or other Post Processor output.

How to use Lua Functions:

Below will describe how you can set up your own Lua Scripts in the Post Processor. The built-in Lua Functions in this document are already defined. You will be able to call these functions two ways:

## Call Lua Function Method 1:

Use **lua\_func\_FunctionName** in a standard Post Block in the Post Processor.

## Call Lua Function Method 2:

Call these Lua functions directly in Lua Blocks 2701 – 2799

* Example Function with No Arguments: **ShowOperationData()**
* Example Function with Arguments: **round(-0.23456, 3)**
* Example Function with Table as Argument: **formatNumber({num = "MILL\_GetXRapid", prefix = "X", numDecimalPlaces = 1 , includeDotAfterInt = false})**

For more details about BobCAD Lua APIs, navigate [**HERE**](https://bobcad.com/components/webhelp/BC_Lua/PostProcessing.html)

For more information about the Lua Programming Language, navigate [**HERE**](https://www.lua.org/manual/5.2/)

## Creating Lua Functions

We have defined two different methods for utilizing Lua scripting with the BobCAD post processing system:

## Method 1

Much like the VB scripting blocks that have existed for years, the same structure can be utilized starting with blocks **2701-2799** within a \*.bcpst post processor file.

By utilizing a variable named **lua\_block\_#** this will call the corresponding block within the post containing the Lua scripting, for example:

lua\_block\_1 calls the 2701. block

lua\_block\_2 calls the 2702. block

…

lua\_block\_99 calls the 2799. block

Method 2

Alternatively, we have created a second method for linking your Lua scripts to a post processor which involves just keeping all your Lua scripts in a separate \*.lua file with the same name as the \*.BcPst file.

Let's say you are working on the BC\_3x\_Mill.bcpst post processor.

## Method 2a

* You can simply create a BC\_3x\_Mill.lua file and put all of your Lua scripts in this file. These files must be located in the same directory, which is typically the Posts\Mill, Posts\Lathe, or Posts\MillTurn folder located in your products Data folder.

**OR**

## Method 2b

You can also create a sub directory inside of your Posts\Mill, Posts\Lathe, or Posts\MillTurn folder and add block 732. Lua sub folder? "MySubFolder" into the post processor itself, and then the posting engine will look for this sub folder and load any \*.lua files that are located inside of this sub folder.

* Inside of your Posts/Mill folder you can create a folder named whatever you want, let's use 3x\_Mill as our folder name.
* Inside of this 3x\_Mill folder you create the MyLua.lua file which contains all of your Lua scripting functions. The name can be whatever you like and you can also have multiple Lua files.
* Add block 732. Lua sub folder? "3x\_Mill" to the BC\_3x\_Mill.bcpst so that the system knows where to find your Lua scripts.

By utilizing a variable named **lua\_func\_FunctionName** inside the post processor this will call the corresponding Lua function from the loaded Lua file using the methods mentioned above.

# BC\_Lua\_Functions.lua

\*\* Use **lua\_func\_** if calling from a standard Post Block. If used in Lua Blocks 2701 – 2799, remove this.

\*\* functions in camelCase are general functions. functions in PascalCase contain BobCAD Lua API functions inside.

|  |  |
| --- | --- |
| **lua\_func\_ShowOperationData** | |
| Mill Job | This function retrieves the current operations ID value of the Operation in the CAM Tree, iterates through the returned table and displays the key-value pairs in a message box.  Use ShowValueFromOperation function to extract the key or subkey you need to use. |
| Lathe Job |
| Mill Turn Job |

|  |  |
| --- | --- |
| **lua\_func\_ShowValueFromOperation("operation\_value")** | |
| Mill Job | This function takes a operation\_value as an argument, retrieves the current operations ID value in the CAM Tree, iterates through the returned table and returns the value of the key or subkey that matches the operation\_value. If the key or subkey is not found, it returns nil.  Use ShowOperationData function to find out to key you need to use |
| Lathe Job |
| Mill Turn Job |

# BC\_NC\_Output\_Lua\_Functions.lua

\*\* Use **lua\_func\_** if calling from a standard Post Block. If used in Lua Blocks 2701 – 2799, remove this.

\*\* functions in camelCase are general functions. functions in PascalCase contain BobCAD Lua API functions inside.

|  |  |
| --- | --- |
| **lua\_func\_round(num, numDecimalPlaces)** | |
| Mill Job | Round a number to a specified number of decimal places |
| Lathe Job | **Args:**  num: The number to be rounded  numDecimalPlaces: The number of decimal places to round to |
| Mill Turn Job | **Returns:**  The rounded number |

|  |  |
| --- | --- |
| **lua\_func\_formatNumber(args)** | |
| Mill Job | Format a number to a specified number of decimal places, with optional leading zero, thousands separator, and dot after integer. |
| Lathe Job | **Args:** A table with the following keys:  **num**: (required) The number to be formatted  **numDecimalPlaces**: The number of decimal places to round to.  Default is rounded to 4 decimal places.  **multiply**: (Optional) A multiplier to be applied to the number before formatting.  Default is 1.  **add**: (Optional) A value to be added to the number before formatting. Default is 0.  **subtract**: (Optional) A value to be subtracted from the number before formatting.  Default is 0.  **divide**: (Optional) A value to be divided by the number before formatting.  Default is 1.  **includeLeadingZero**: (Optional) Whether to include a leading zero for numbers less  than 1. Default is true.  **useThousandsSeparator**: (Optional) Whether to include a thousands separator.  Default is false.  **includeDotAfterInt:** (Optional) Whether to include a dot after the integer part if  the number is a whole number. Default is true.  **prefix:** (Optional) A prefix to be added to the formatted number. (Primarily used  for BobCAD API functions) |
| Mill Turn Job | **Returns:**  The formatted number as a string |

formatNumber is a very versatile function that allows you to format just about any value you can think of in a BobCAD-CAM Post Processor.

Using a Table (eg. myTable = { arg1 = 1, arg2 = false }) as an input argument allows you to explicitly define the parameters you want to change. If not explicitly defined, the function will use the default state.

Example of formatNumber:

There are 3 ways you can input a value:

**Method 1:** input any number for the “num” argument

lua\_func\_formatNumber({num = 24, includeDotAfterInt = false}) // Ouput: 24

**Method 2:** input a string of a VBScript BobCAD API with no () for the “num” argument

(If Lathe Job X Rapid plane is 2.25 (radius))

lua\_func\_formatNumber({num = "MILL\_GetXRapid", prefix = "X", numDecimalPlaces = 1 , includeDotAfterInt = false}) // Outputs: X2.3

**Method 3:** Use the BcPost.RunVBApi(“VBScript\_BobCAD\_API”) function to grab any value and

use for the “num” argument

lua\_func\_formatNumber({num = BcPost.RunVBApi("MILL\_GetXRapid") , prefix = "X", numDecimalPlaces = 1 , includeDotAfterInt = false}) // Outputs: X2.3

**Note:** You can use method 3 to obtain the value of any VBScript BobCAD API and use it as an input for any Lua Function if needed.

|  |  |
| --- | --- |
| **lua\_func\_convertAngle(angle, mode, numDecimalPlaces)** | |
| Mill Job | Convert an angle between degrees and radians. |
| Lathe Job | **Args:**  angle: The angle to be converted  mode: The conversion mode. Can be "degreesToRadians" or "radiansToDegrees".  numDecimalPlaces: (Optional) The number of decimal places to round to.  Default is no rounding. |
| Mill Turn Job | **Returns:**  The converted angle |

Example of convertAngle:

lua\_func\_convertAngle(180, "degreesToRadians") // Outputs: 3.14159265358979

lua\_func\_convertAngle(180, "degreesToRadians", 4) // Outputs: 3.1416

lua\_func\_convertAngle(3.1415, "radiansToDegrees", 0) // Outputs: 180

**Note:** input args sequentially if args is not a table. (eg. lua\_func\_convertAngle(3.1415, 0) is invalid)

|  |  |
| --- | --- |
| **includeDotAfterNum(num, includeDotAfterInt)** | |
| Mill Job | Include a dot after the integer numbers. Ideally used in lua blocks 2701 -2799. |
| Lathe Job | **Args:**  num: The number to be formatted  includeDotAfterInt: (true or false) Whether to include a dot after the integer part  of the number |
| Mill Turn Job | **Returns:**  The number with a dot after if it is an integer and if includeDotAfterInt is true |

|  |  |
| --- | --- |
| **GetValueFromOperation(“operation\_value”)** | |
| Mill Job | Get a value from the current operation based on a search key. Use lua\_func\_ShowOperationData to figure out the “operation\_value”. Ideally used in lua blocks 2701 -2799. |
| Lathe Job | **Args:**  “operation\_value”: The key to search for in the operation's parameters. |
| Mill Turn Job | **Returns:**  The value associated with the search key, or nil if the key is not found. |

Example GetValueFromOperation:

-- Get the thread pitch from the current operation

local threadPitch = GetValueFromOperation("thread\_pitch")

-- Outputs: The value of thread\_pitch, or nil if not found

## General Functions

|  |  |
| --- | --- |
| **lua\_func\_UnitsComment** | |
| Mill Job | Outputs a comment in the NC file with the units of the job. |
| Lathe Job | **Args:**  “operation\_value”: The key to search for in the operation's parameters. |
| Mill Turn Job | **Returns:**  Outputs a comment in the NC file with the units of the job. |
| **Used for Post Blocks:**  Used for start of file blocks, but could also be used in tool change blocks as well  630 and 631: Adjust these post questions to set the comment syntax | |

|  |  |
| --- | --- |
| **lua\_func\_IfDwellOutput(prefix, includeDotAfterInt)** | |
| Mill Job | Check if a dwell exists and ouput with a prefix if it does. This function is used for Peck Drilling cycles since they do not have a separate dwell post block. |
| Lathe Job | **Args:**  prefix: The prefix to be used in the dwell value  includeDotAfterInt: (true or false) Whether to include a dot after the integer part  of the value |
| Mill Turn Job | **Returns:**  The dwell value with a prefix if dwell exists, otherwise nil. |
| **Used for Post Blocks:**  Mill:  73. High speed peck drill canned cycle - Fast peck  83. Peck drill canned cycle  Any other post block that uses a 'dwell' post variable  Lathe:  1126. Peck drill canned cycle  1121. High speed peck drill canned cycle  Any other post block that uses a 'dwell' post variable | |

## Lathe Functions

|  |  |
| --- | --- |
| **lua\_func\_ThreadsPerInch(prefix, numDecimalPlaces, includeDotAfterInt)** | |
| Lathe Job | Convert a pitch value to threads per inch for the Lathe Thread Operation. |
| Mill Turn Job | **Args:**  prefix: The prefix to be used in the threads per inch value  numDecimalPlaces: (Optional) The number of decimal places to round the threads per  inch value to. Default is 4.  includeDotAfterInt: (Optional) (true or false) Whether to include a dot after the integer  part of the value |
|  | **Returns:**  The threads per inch value with a prefix rounded to the nearest whole number. |
| **Used for Post Blocks:**  1087 (Start of thread (G76) cycle) | |

|  |  |
| --- | --- |
| **lua\_func\_RadiusIArcMoveBlock1025** | |
| Lathe Job | Outputs the Lathe arc move post block with I values converted from diameter to radius. |
| Mill Turn Job | **Args:**  none |
|  | **Returns:**  The whole post block for the Lathe Arc Move (Post Block: 1025) with I values  converted from diameter to radius. |
| **Used for Post Blocks:**  1025 (Arc move (Lathe)) | |

|  |  |
| --- | --- |
| **lua\_func\_ArcCenterXToRadius(prefixI, prefixK, numDecimalPlaces)** | |
| Lathe Job | Outputs the arc center I and K (Or other specified prefix) values for the Lathe Arc Move post block. |
| Mill Turn Job | **Args:**  prefixI: The prefix to be used in the I value  prefixK: The prefix to be used in the K value  numDecimalPlaces: (Optional) The number of decimal places to round the I and K  values to. Default is 4. |
|  | **Returns:**  The I and K (Or other specified prefix) values for the Lathe Arc Move post block. |
| **Used for Post Blocks:**  1025 (Arc move (Lathe)) | |

Example ArcCenterXToRadius:

Use lua\_func\_ArcCenterXToRadius("I", "K", 4)

Example Post Block Line: n,g\_arc\_move,x\_f,z\_f,lua\_func\_ArcCenterXToRadius("I", "K", 4),feed\_rate

# BC\_Adv\_Posting\_Page.lua

\*\* Use **lua\_func\_** if calling from a standard Post Block. If used in Lua Blocks 2701 – 2799, remove this.

\*\* functions in camelCase are general functions. functions in PascalCase contain BobCAD Lua API functions inside.

The following functions allow you to define an Adv Posting Page directly in the Post Processor without having to create the adv posting custom file manually.

It is REQUIRED to have the **lua\_func\_FinalizeAdvPostingPage** function at the end of the Create functions to create the Adv Posting Custom File.

View the Advanced Posting with Custom Files link [**HERE**](https://bobcad.com/components/webhelp/PostProcessorHelpSystemFiles/Topics/advancedpostingwithcustomfiles1.html) for more info about the Adv Posting Page

### Example Initialization of the Advanced Posting Page in the “Current Settings” page of a Job:

0. File Header

// Initialize the Adv Posting Page

lua\_func\_CreateCheckBox({setPosition = 1, assignCheckBoxLabel = "Use Tool Changer", setDefaultToOnOff = 1})

lua\_func\_CreateComboBox({setPosition = 1, assignComboBoxLabel = "ComboBox Label 1", setDefaultSelection = 1, assignChoiceLabels = {"Choice 1", "Choice 2", "Choice 3"}})

lua\_func\_CreateIntegerEditBox({setPosition = 5, assignEditBoxLabel = "Integer Edit Box", setDefaultIntegerNumber = 10})

lua\_func\_CreateRealEditBox({setPosition = 2, assignEditBoxLabel = "Real Edit Box", setDefaultDecimalNumber = 1.23})

lua\_func\_CreateStringEditBox({setPosition = 3, assignEditBoxLabel = "String Edit Box", setDefaultStringText = "Default Text"})

lua\_func\_FinalizeAdvPostingPage({postProcessorName = "BC\_Single\_Line\_TESTING\_LUA\_FUNCS", extension = "CustomSettings", jobType = "Lathe"})

A screenshot of a computer

Description automatically generated

Below is an image of all the possible combinations of the Adv Posting page:

A screenshot of a computer

Description automatically generated

|  |  |
| --- | --- |
| **lua\_func\_FinalizeAdvPostingPage(args)** | |
| Mill Job | Takes the string text of the Create functions defined and creates an Advanced Posting Custom File in the specified C:\BobCAD-CAM Data\BobCAD-CAM V36\Posts folder. |
| Lathe Job | **Args:** A table with the following keys:  postProcessorName: A string of the exact name of the post processor minus the  extension.  extension: A string of the extension of the Advanced Posting Custom file. Click this  [**LINK**](https://bobcad.com/components/webhelp/PostProcessorHelpSystemFiles/Topics/advancedpostingwithcustomfiles1.html) to view the different extensions used.  jobType: A string of the post processor’s job type.  Use: “mill”, “lathe”, or “millturn” |
| Mill Turn Job | **Returns:**  An Advanced Posting Custom File place in the BobCAD-CAM Data folder. View the  “Adv Posting” page in the software once you “Post” out the current job at least  once. |
| **Used for Post Blocks:**  Ideally, use in the 0. File Header | |

Example FinalizeAdvPostingPage:

lua\_func\_FinalizeAdvPostingPage({postProcessorName = "BC\_Single\_Line\_TESTING\_LUA\_FUNCS", extension = "CustomSettings", jobType = "Lathe"})

**IMPORTANT:** This function is require for all post processors that utilize the Adv Posting Lua Functions. Place it at the bottom of the all the Create Adv Posting lua functions.

|  |  |
| --- | --- |
| **lua\_func\_CreateCheckBox(args)** | |
| Mill Job | Creates a Check Box on the Adv Posting Page set to a default value. |
| Lathe Job | **Args:** A table with the following keys:  setPosition: An integer number that sets the location of the check box on the Adv  Posting page. Starting at Position 1 through 9  assignCheckBoxLabel: A string of the name given to the check box to distinguish  what the check box is used for.  setDefaultToOnOff: An integer number that sets the default value of the check  box. (0 = Off, 1 = On) |
| Mill Turn Job |  |
| **Used for Post Blocks:**  Ideally, use in the 0. File Header | |

Example CreateCheckBox:

lua\_func\_CreateCheckBox({setPosition = 1, assignCheckBoxLabel = "Use Tool Changer", setDefaultToOnOff = 1})

A screenshot of a computer

Description automatically generated

|  |  |
| --- | --- |
| **lua\_func\_CreateComboBox(args)** | |
| Mill Job | Creates a Combo Box on the Adv Posting Page set to a default value. |
| Lathe Job | **Args:** A table with the following keys:  setPosition: An integer number that sets the location of the check box on the Adv  Posting page. Starting at Position 1 through 25. Positions 21 – 25 are  wider.  assignComboBoxLabel: A string of the name given to the combo box to distinguish  what the combo box is used for.  setDefaultSelection: An integer number that sets the default choice of the combo  box. Starting at an index of 0 through the number of choice  labels setup. (eg. Choice 1 = 0, Choice 3 = 2)  assignChoiceLabels: a table of choices for selection in the combo box. eg.  assignChoiceLabels = {"Choice 1", "Choice 2", "Choice 3"} |
| Mill Turn Job |  |
| **Used for Post Blocks:**  Ideally, use in the 0. File Header | |

Example CreateComboBox:

lua\_func\_CreateComboBox({setPosition = 1, assignComboBoxLabel = "ComboBox Label 1", setDefaultSelection = 1, assignChoiceLabels = {"Choice 1", "Choice 2", "Choice 3"}})

A close-up of a computer screen

Description automatically generated

|  |  |
| --- | --- |
| **lua\_func\_CreateIntegerEditBox(args)** | |
| Mill Job | Creates an Integer Edit Box on the Adv Posting Page set to a default value. |
| Lathe Job | **Args:** A table with the following keys:  setPosition: An integer number that sets the location of the check box on the Adv  Posting page. Starting at Position 1 through 25. Positions 21 – 25 are  wider.  assignEditBoxLabel: A string of the name given to the integer edit box to  distinguish what the edit box is used for.  setDefaultIntegerNumber: An integer number that sets the default edit box value. |
| Mill Turn Job |  |
| **Used for Post Blocks:**  Ideally use in the 0. File Header | |

Example CreateIntegerEditBox:

lua\_func\_CreateIntegerEditBox({setPosition = 5, assignEditBoxLabel = "Integer Edit Box", setDefaultIntegerNumber = 10})

A screenshot of a computer

Description automatically generated

|  |  |
| --- | --- |
| **lua\_func\_CreateRealEditBox(args)** | |
| Mill Job | Creates a Real Edit Box on the Adv Posting Page set to a default value. |
| Lathe Job | **Args:** A table with the following keys:  setPosition: An integer number that sets the location of the check box on the Adv  Posting page. Starting at Position 1 through 25. Positions 21 – 25 are  wider.  assignEditBoxLabel: A string of the name given to the real edit box to  distinguish what the edit box is used for.  setDefaultDecimalNumber: A decimal number that sets the default edit box value. |
| Mill Turn Job |  |
| **Used for Post Blocks:**  Ideally, use in the 0. File Header | |

Example CreateRealEditBox:

lua\_func\_CreateRealEditBox({setPosition = 2, assignEditBoxLabel = "Real Edit Box", setDefaultDecimalNumber = 1.23})

A screenshot of a computer

Description automatically generated

|  |  |
| --- | --- |
| **lua\_func\_CreateStringEditBox(args)** | |
| Mill Job | Creates a Real Edit Box on the Adv Posting Page set to a default value. |
| Lathe Job | **Args:** A table with the following keys:  setPosition: An integer number that sets the location of the check box on the Adv  Posting page. Starting at Position 1 through 25. Positions 21 – 25 are  wider.  assignEditBoxLabel: A string of the name given to the string edit box to  distinguish what the edit box is used for.  setDefaultDecimalNumber: A string that sets the default edit box value. |
| Mill Turn Job |  |
| **Used for Post Blocks:**  Ideally, use in the 0. File Header | |

Example CreateStringEditBox:

lua\_func\_CreateStringEditBox({setPosition = 3, assignEditBoxLabel = "String Edit Box", setDefaultStringText = "Default Text"})

A screenshot of a computer

Description automatically generated

# BobCAD Lua APIs – Post Processing

|  |  |
| --- | --- |
| **BcPost.RunVBApi("VBScript\_API\_Name")** | |
| Mill Job | Turn off automatic debug comments in post output |
| Lathe Job |
| Mill Turn Job |