

# **BcadTools Freeware Edition**

# Introduction

---



## BcadTools - Freeware utility for the Bricscad system

This software is intended to provide some rarely used AutoCAD® commands for Bricscad, which are not available otherwise, in order to improve Bricscad comfort and compatibility for both users and applications.

'BcadTools Freeware' also adds a number of common-purpose functions and commands to provide more comfort for the usual daily work - for example, the **LayerTools** feature provides very powerful means to manage drawing layers.

As from initial publishing, the development will be continued - more features and functionality will be added, implementing bugfixes, user ideas and wishes as well. We will also keep focus on compatibility issues and improvements for Bricscad, especially for application code.

For a detailed description of all functions and commands, please refer to chapter [Commands and Functions](#).



# License

---

## BcadTools Freeware Edition - License Agreement

By using, copying or distributing BcadTools and its documentation (the application code and documentation are collectively referred to as „the software“), you indicate your acceptance of the following license terms :

### 1. Usage

Except where otherwise noted, all of the documentation and software included in the BcadTools package is copyrighted by TM-CAD Engineering.

TM-CAD Engineering grants you (the licensee) the permission to use the software without any cost, for both commercial and non-commercial purposes on any computer in your possession. You are also allowed to make copies and distribute the software according to requirements of section 2 below.

### 2. Distribution

As used in this License Agreement, the term "distribute" (and its variants) includes making the Software available (either by intention or unintentionally) to third parties for copying or use. For any distributed copy of this software, you may not impose any further restrictions on the recipients' exercise of the rights granted herein.

You may distribute the software under the following conditions :

- You may not modify, combine other commercial applications with, or otherwise prepare derivative works of the software.
- You may not distribute the software for a fee (by payment of money or otherwise, whether direct or indirect). In particular (but not limited to), you may not charge for the software to recover media costs associated with its distribution, you may not charge for a product or compilation

- the software is a part of, you may not provide a paid download of the software and you may not require a registration of some kind (for example, on your website) for downloading the software.
- You must include this Licence Agreement in original form on all copies of the software distributed.
  - You must not distribute the software in a manner, which provides the impression that this software is not Freeware, but involves fees for its usage.

As an exception to the condition above,

- You may personally install this software as part of a free or paid commercial service.
- You may distribute this software along with computer hardware.

### **3. Warranties and Disclaimer**

The software and any support from TM-CAD Engineering are provided "AS IS" and without any warranty, express or implied.

TM-CAD Engineering specifically disclaim any implied warranties of merchantability and correctness for a particular purpose. In no case TM-CAD Engineering will be liable for any damages, including and not limited to any lost profits, lost savings or any incidental or consequential damages, whether resulting from impaired or lost data, software or computer failure or any other cause, or for any other claim by the user or for any third party claim.

### **4. Limit of Liability**

Subject to any liability which may not be excluded or limited by law the licensor shall not be liable and hereby expressly excludes all liability for loss or damage howsoever and whenever caused to you.

### **5. Termination**

The rights granted to you under this licence shall terminate automatically upon any breach by the licensee of the terms of this Licence Agreement. Individuals or entities who have received the software from you under this licence, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses.



## **Installation**

---

### **Installing BcadTools Freeware**

We provide 2 different installation sets to get BcadTools Freeware installed - a normal [Installer package](#), and also a [ZIP archive](#).

Both is available for download from website [www.LT-Extender.com](http://www.LT-Extender.com).

Before installing BcadTools Freeware, please ensure that Bricscad is not running.

#### **using Installer package**

The normal installer package only asks for a disk location where to install BcadTools Freeware. It also adds a few settings to Bricscad Registry to allow an automatic loading and command registration for BcadTools.

#### **using ZIP archive**

To install using the ZIP archive, please follow these simple steps :

- unzip as a separate folder, to any suitable location as preferred
- start Bricscad
- use **\_APpload** command and load the file **BcadTools.10.brx** file from that location

»»» this is all - nothing more to be done.

---

When BcadTools Freeware is initially loaded, it shows a command line message

**"BcadTools Freeware Edition - (c)2010 TM-CAD Engineering Torsten Moses"**

and also loads its menu and completes "DemandLoad" Registry entries.

In further Bricscad sessions, this BcadTools message might not be visible at Bricscad startup - this is normal, because BcadTools is only loaded when a BcadTools command is started; this behaviour is called "DemandLoad".

#### **Remarks :**

BcadTools Freeware does not need any extra support paths to be defined.

At further Bricscad starts, BcadTools application file is initially loaded when one of the BcadTools commands is started.



## Start with Bricscad

---

### **Running BcadTools after Installation**

After the installation, at next start of Bricscad, the BcadTools Freeware is loaded automatically - the menu is also loaded, and you will see a message in command line window :

**"BcadTools Freeware Edition - (c)2010 TM-CAD Engineering Torsten Moses"**

**Then, everything is fine.**

**Only, if you do not see that command line message - or after a manual installation** (using ZIP install package) - then there is a very simple way to activate BcadTools - please follow these instructions :

1. start Bricscad
2. use **\_APpload** command and load the file **BcadTools.10.brx**  
(you can also use Drag & Drop, and drop the **BcadTools.10.brx** file into Bricscad command line)

These manual steps need to be processed only once.

#### **Remarks :**

**In further Bricscad sessions**, this BcadTools message might not be visible at Bricscad startup - this is normal, because BcadTools is only loaded when a BcadTools command is started; this behaviour is called "DemandLoad".



## Software-Updates

---

### Installing Updates for BcadTools Freeware

To update your BcadTools Freeware installation with a newer version, you can use the normal installation procedure.

We provide 2 different installation sets to get BcadTools Freeware installed - a normal [Installer package](#), and also a [ZIP archive](#).

Both is available for download from website [www.LT-Extender.com](http://www.LT-Extender.com).

Before updating BcadTools Freeware, please ensure that Bricscad is not running.

#### using Installer package

Update procedure is very easy - simply install into existing BcadTools Freeware folder on disk. Normally, the Installer package remembers that disk folder and suggests to install there. However, you can run Bricscad and open BcadTools Info (About) box, which also shows the location of BcadTools Freeware on disk.

#### using ZIP archive

To update using the ZIP archive, please unzip into existing BcadTools Freeware folder.

»»» this is all - nothing more to be done.

---

#### Remarks :

In any case, you can always install or unzip into existing BcadTools Freeware folder on disk ! There is no risk for any kind of damage or loss of user-specific settings at all.



---

## Uninstallation

### Uninstallation of BcadTools Freeware

As for any normal Windows application, BcadTools Freeware can be uninstalled using ControlPanel->Software, or using the Startmenu program group, which also contains the uninstallation entry.

#### manual uninstallation :

If there is something broken, and this automatic uninstallation will fail, you can manually uninstall BcadTools.

- delete the BcadTools program folder from disk
- these Registry keys can be deleted using regedit.exe :
  - HKEY\_CURRENT\_USER\Software\TM-CAD Engineering\BcadTools
  - HKEY\_CURRENT\_USER\Software\Bricsys\ObjectDRX\V10\Applications\BcadTools
  - HKEY\_CURRENT\_USER\Software\Bricsys\Bricscad\V10\en\_US\Applications\BcadTools

These Registry keys **V10** and **en\_EN** are version and language dependent - for other Bricscad versions and language packages, please look at the appropriate version and language keys, and delete BcadTools key at those locations.

Even if these Registry keys are not deleted and remain orphaned - they do not effect correct operation of Bricscad.



# Using BcadTools

---

## User Interface and Commands

BcadTools Freeware uses a normal pull-down menu and both command line and dialog based commands.

### User Interface language

BcadTools is installed with a number of languages (minimum : English and German). There is an automatism that determines which language is used for the menu, the dialogs and the command messages (for details see [Translations](#)) - normally, BcadTools will try to use the same language as Bricscad itself.

It is also possible to force BcadTools to use any other installed language - please see [Interface Language](#).

### Command names

All commands are available with (internal) English names as well as with localized names (if translation is available). So this documentation will refer to English command names, with "\_" prefix. Also, the menu itself uses English command names, using the "\_" prefix.

### Tooltips

Some dialogs show tooltips to display hints regarding dialog elements - here, we use an extended tooltip window, which is able to show a normal and (alternatively) a long tooltip text.

All \*NOT\* selected layers will be used▶

◀Using this option allows do process those layers, which are \*NOT\* selected - i.e. the usual "Isolate" function is available as "Off" using \*NOT\* selected layers, thus, only the selected layer(s) remain visible.  
This option is respected by "Off", "Freeze", "VP-Freeze", "Lock" and "Unlock"

The tooltip window opens with the normal hint text - if there is an extended hint available, a small black triangle is shown at the right edge.

You can move the mouse pointer into the tooltip window - it will show the extended hint text, if available; moving the mouse pointer outside the tooltip window will switch back to the normal hint text.

The number of tooltips using this extended hint text will be increased by continued development.

### Online help and Documentation

We provide online help and documentation in 3 formats :

1. as compiled HTMLHelp file (\*.chm)
2. as standard HTML web page (\*.htm)
3. as PDF file (\*.pdf)

When online help is requested from BcadTools Freeware, the program tries to use the several help files in above sequence, until the help file can be found and opened.

The number of dialogs providing context-specific help access will be increased by continued

development.

### Remarks regarding CHM help access in Windows 7

Unfortunately, Windows 7 does no longer install the HTMLHelp runtime system (to display CHM files) by default.

In this case, or whenever you encounter a problem to display CHM help file, you can download and install the latest version of HTMLHelp from Microsoft using these download links :

[HTMLHelp download #1](#)

[HTMLHelp download #2](#)

After download, please run the installer.

## Commands

All commands and functions are available from pull-down menu, and most commands are also available from a toolbar as well.

Some commands and functions additionally provide a programming interface (Lisp), these can be called as Lisp expression also. In generally, and if not otherwise specified, all commands and functions can be called via (command) resp. `acedCommand()` API.

The number of commands available will be increased by continued development.

For a complete description of all implemented functions please see chapter [Commands and Functions](#)

.



## Interface Language

---

### Select User Interface Language

Command: `bcadtools_language`

BcadTools Freeware allows to override default, automatic user interface language selection. The pull down menu provides an entry to start this language selection dialog.



#### Automatic

using this option configures BcadTools to use the same language as Bricscad (or English, if that language is not available)

#### Popup Listbox

if **Automatic** option is unchecked, this listbox shows all available language - to select the desired one.

Any change of user interface language has immediate effect - the appropriate menu is loaded, and all dialog and command messages are taken from selected language files.

This configuration is user-dependent and stored in Windows Registry under

**HKEY\_CURRENT\_USER\Software\TM-CAD Engineering\BcadTools** - under key **Language**.



## Commands and Functions

---

### Commands Overview

Here is an alphabetical list of all commands, with a short description.

 <a href="#">Arx</a>	Loads, Unloads, Lists Brx/Drx applications, also lists commands of loaded applications
 <a href="#">AttReDef</a>	Allows to redefine a block definition incl. contained attribute definitions
 <a href="#">AttSync</a>	Updates the attributes of all instances of a specified block with current attribute definitions
 <a href="#">Battman</a>	Edits the attributes of a block definition; optionally also updates existing block references
 <a href="#">Battedit</a>	Edits the attributes of a block reference; like "EAttEdit", but also allows to edit the "Visible" flag
 <a href="#">Compile</a>	Compiles SHP shape files into SHX files
 <a href="#">Dxbln</a>	Imports DXB (Drawing eXchange Binary) encoded binary files into the drawing
 <a href="#">LayerTools</a>	Dialog to manage drawing layers based on contained entities, incl. reverse selection and restore
 <a href="#">MLine</a>	Draws MultiLine entities
 <a href="#">MLStyle</a>	Manages MultiLine styles
 <a href="#">MLEdit</a>	Edit MultiLine vertices, cuts / breaks, and intersections
 <a href="#">MLModi</a>	Edit MultiLine properties



## Arx

---

### Loads, Unloads, Lists Brx/Drx applications, also lists the commands of loaded application

 Command: **Arx**  
Enter an option [[?/Load/Unload/Commands](#)]:

#### ? - lists loaded applications

Lists all loaded applications, which are based on BRX/DRX C++ interface; these can be Bricscad-internal or external third-party applications; typical file extensions are \*.arx, \*.brx, \*.dbx, \*.drx.

#### Load - loads a specified application

Opens the file dialog to select a BRX/DRX based application, or queries for the application filename at command line (depending on context); if the filename selected is valid, the application will be loaded.

#### Unload - unloads a specified application

Unloads the specified BRX/DRX application.

## Commands - lists registered commands

Lists all commands registered by BRX/DRX applications and modules



## AttRedef

---

### Allows to redefine a block definition incl. contained attribute definitions

 Command: **Attredef**

Enter name of the block you wish to redefine: <enter name of block>

Select objects for new Block ... <select new block entities>

Specify insertion base point of new Block <specify new base point>

Attributes of existing block references retain their current values after block redefinition, if the new block definition contains the same attribute (based on the tag).

Attributes of existing block references which are not contained in new block definition (based on the tag) are deleted from existing references.

New attributes not existing in current block definition will be assigned their default values.



## AttSync

---

### Updates the attributes of all instances of a specified block with current attribute definitions

 Command: **Attsync**

Enter an option [[?](#)/[Name](#)/[Select](#)] <Select>:

#### ? - lists block with attributes

This option lists all block definitions which contain attributes.

#### Name - select by name

Allows to specify the block definition to be synchronized by entering its name.

#### Select - select by entity

This option allows to specify the block definition to be synchronized by selecting a block reference.

---

Attsync command synchronizes attributes of existing block references with the current block definition; attribute values assigned to existing attributes are not changed.

#### Remarks :

because the block reference and attributes are recreated from block definition, any assigned

extended entity data will be lost !



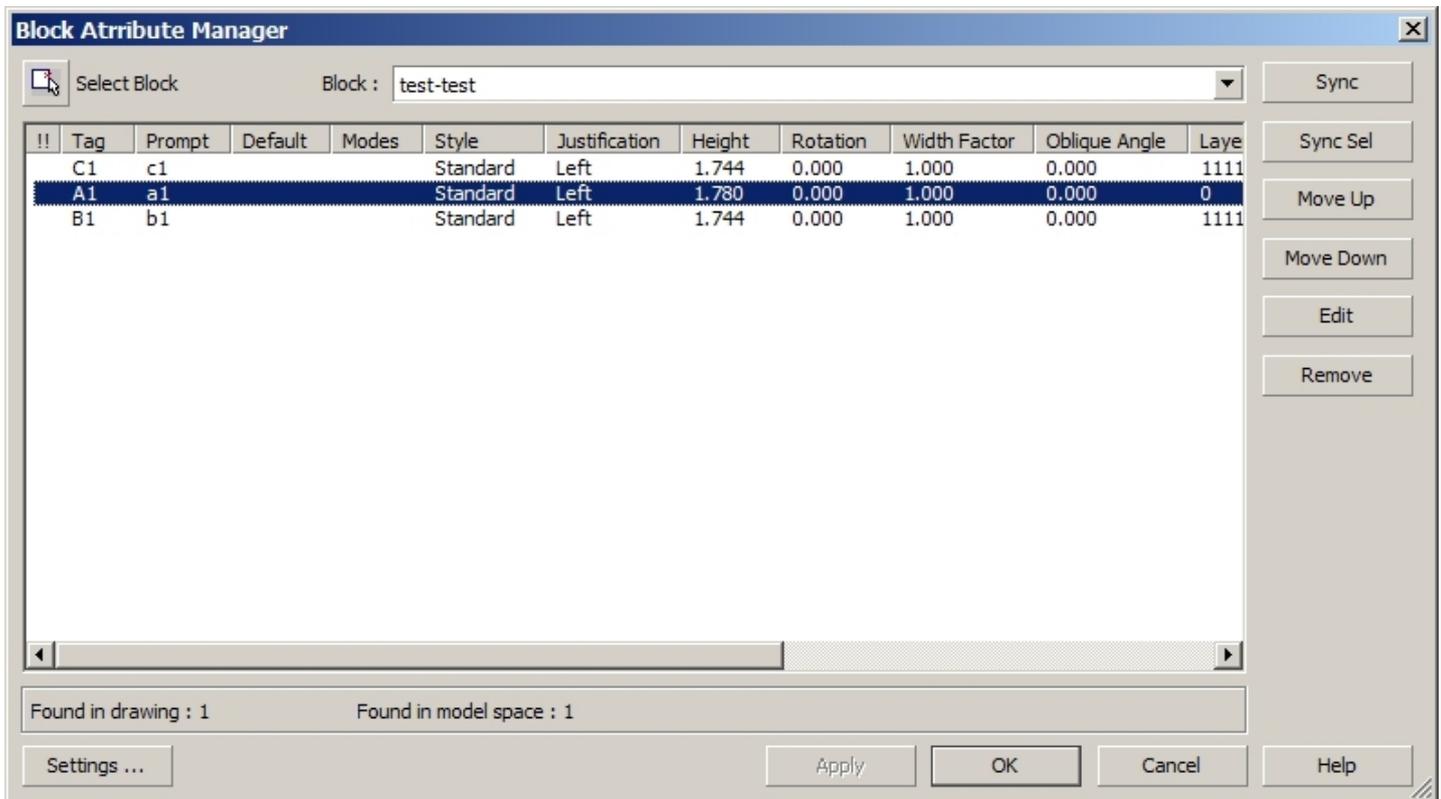
## Battman

### Edits the attributes of a block definition; optionally also updates existing block references

 Command: **Battman**

This command opens the Block Attribute Manager dialog.

If the current drawing does not contain block references with attributes, a message is displayed, and the dialog is not opened.



Block Attribute Manager - main dialog

This dialog is free resizeable, and will maintain size and position of the dialog across Bricscad sessions.

### Select Block

You can select the block definition to be modified using the mouse, and select a block reference - afterwards the block is select.

If the block you select does not contain attributes, the active block definition is not changed in Block popup list.

### Block

The popup list shows all block definitions containing attributes. You can select the block definition to be modified.

## Sync

Updates all block references with current attributes definitions. Attribute values are not changed.

## Sync Sel

Updates selected block references with current attributes definitions. Attribute values are not changed.

## Move Up / Down

Allows to change the attribute's position in prompt sequence (does not effect constant attributes, because these are not prompted for a value)..

## Edit

Opens the [Edit Attribute Dialog](#) to edit the attribute's properties.  
Using left mouse button doubleclick also opens the dialog.

## Remove

Removes the selected attribute from the block definition. If there is only 1 attribute left, it can not be removed from the block definitions.

Using **Apply** will also remove the related attribute(s) from all block references.

If option "Apply changes to existing references" is active in BAttMan settings, the attribute is immediately removed from all references.

## Settings ...

Opens the [Block Attribute Manager Settings Dialog](#) to specify behaviour and list box columns to be displayed.

## Apply

Updates the block definition with current settings. If the option "Apply changes to existing references" is active, then Apply also effects all existing block references.



## Edit Attributes Dialog

---

### Edit the block attribute properties

The dialog always shows the name of current block definition, as well as the name of active attribute, which will be modified.

## Auto Preview Changes

This option is available only, if "Apply changes to existing references" is active in [Block Attribute Manager Settings Dialog](#).

If this option is active, any changes to attribute properties are immediately updated to this attribute block references.

**Mode Invisible** sets the attribute as visible or invisible

**Mode Constant** if active, the attribute uses default value and can't be changed

**Mode Verify** if active, the user will be prompted to confirm attribute value, when a new block reference is inserted

**Mode Preset** if active, the attribute uses the default value

**Tag** specifies the attribute identifier (should be unique !)

**Prompt** the prompt string to be displayed when the block is inserted

**Default** specifies the default value for the attribute (see Mode Preset)

**Text Style** specifies the text style assigned to the attribute

**Justification** specifies the text justification assigned to the attribute

**Height** specifies the text height assigned to the attribute

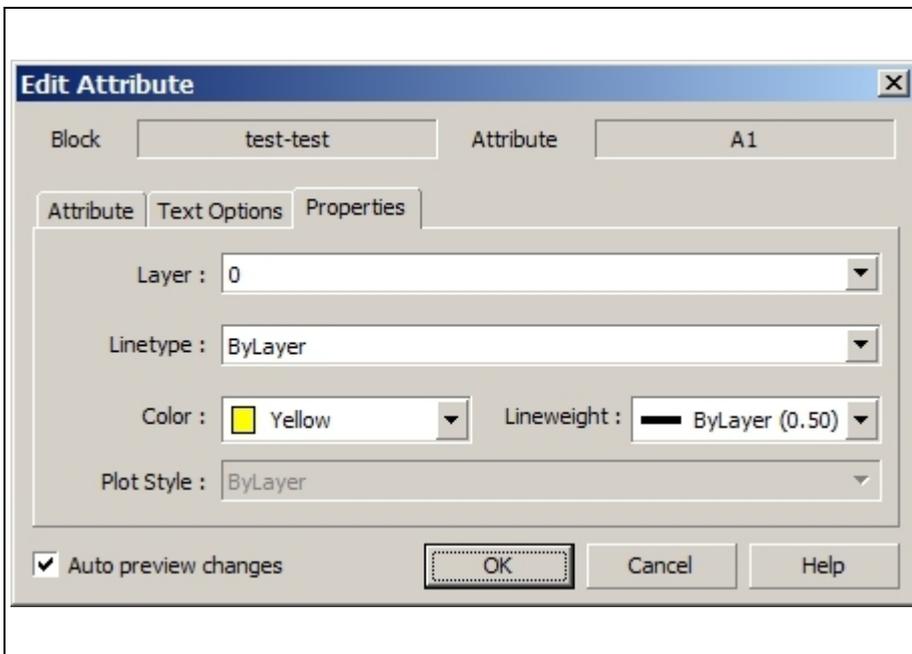
**Rotation** specifies the text rotation angle assigned to the attribute

**Backwards** specifies whether the attribute is displayed backwards

**Upside down** specifies whether the attribute is displayed upside-down

**Width Factor** specifies the text width assigned to the attribute

**Oblique Angle** specifies the text oblique angle assigned to the attribute



**Layer** specifies the layer assigned to the attribute

**Linetype** specifies the linetype assigned to the attribute

**Color** specifies the color assigned to the attribute

**Lineweight** specifies the lineweight assigned to the attribute

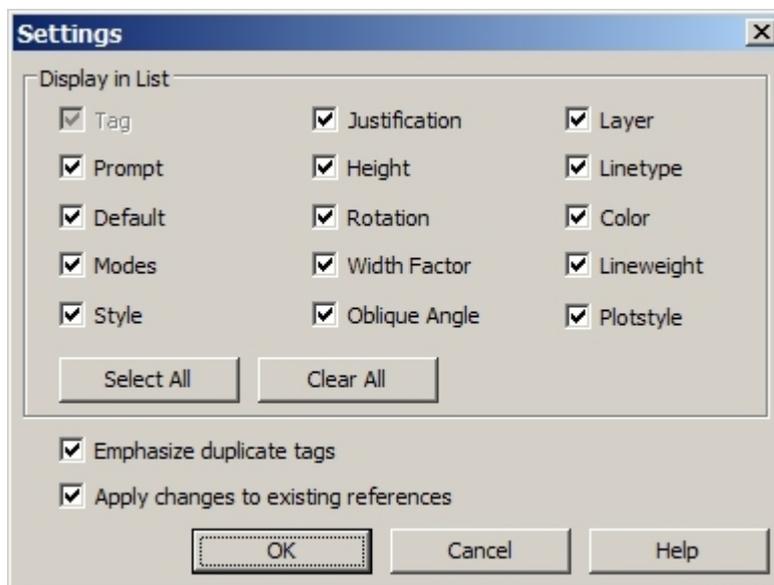
**Plot Style** specifies the plotstyle assigned to the attribute (disabled when the drawing is in color-dependent mode)



## Settings Dialog

### Edits Battman Settings

Specifies the property columns to be displayed for the attributes list..  
It also defines some basic behaviour for the Battman dialog.



settings dialog

### Display in List

This allows to customize which property columns should be displayed for the attribute list - the TAG filed can not be disabled.

These settings are permanent across Bricscad sessions.

## Emphasize duplicate tags

If this option is active, duplicate attribute tags will be emphasized in the attribute list - it is suggested to keep this option active.

## Apply changes to existing references

Specifies whether existing block references shall be updated with new properties. If this option is not active, only attributes for new block references will use new attribute settings. Alternatively, you can use **Sync** or **Sync Sel** functions in main dialog, to update all existing or only selected block references.



## Battedit

### Edits the attributes of a block reference

This command is very similar to Bricscad's **EATTEDIT** command, except that **BATTEDIT** allows to define the visibility of any attribute.

 Command: **Battedit**

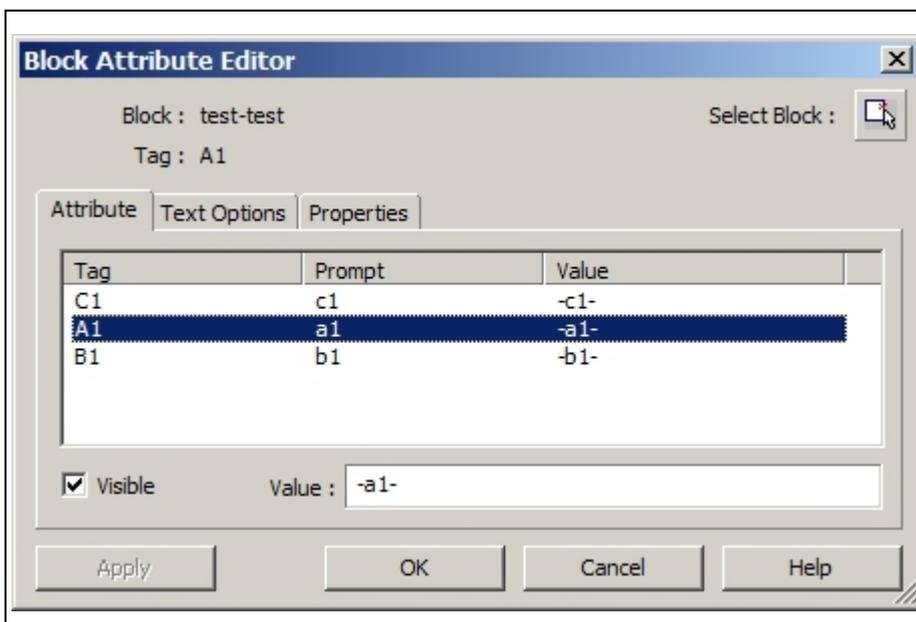
Select a block: <select a block reference>

If you select a block reference containing attributes, the Battedit dialog is opened. If the block reference does not contain attributes, or if you select anything other than a block reference, an error message is shown, and selection is repeated.

The dialog always shows the name of current block definition, as well as the name of active attribute, which will be modified.

## Apply

If an attribute property has been changed, the Apply button gets active - if used, it updates the block reference with current attribute values.

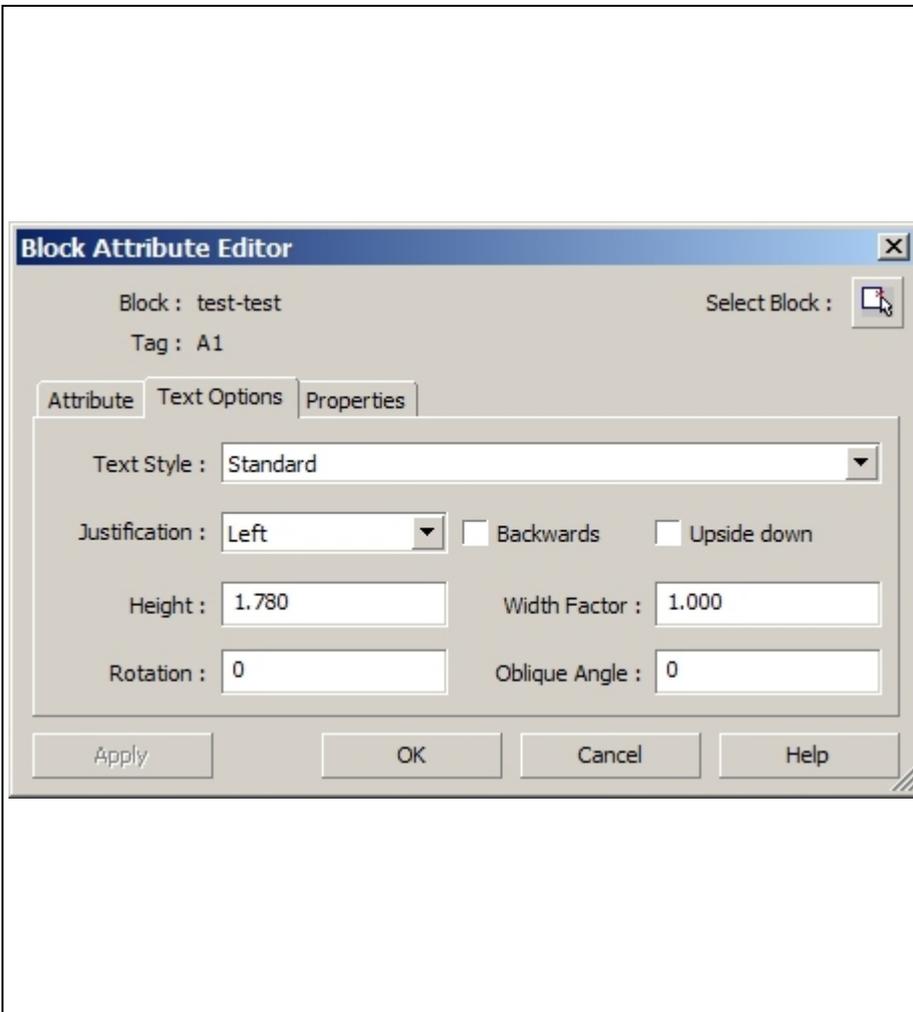


**Tag** specifies the attribute identifier (should be unique !)

**Prompt** the prompt string to be displayed when the block is inserted

**Value** specifies the current value for the attribute

**Visible** specifies whether the attribute is visible or not



**Text Style** specifies the text style assigned to the attribute

**Justification** specifies the text justification assigned to the attribute

**Height** specifies the text height assigned to the attribute

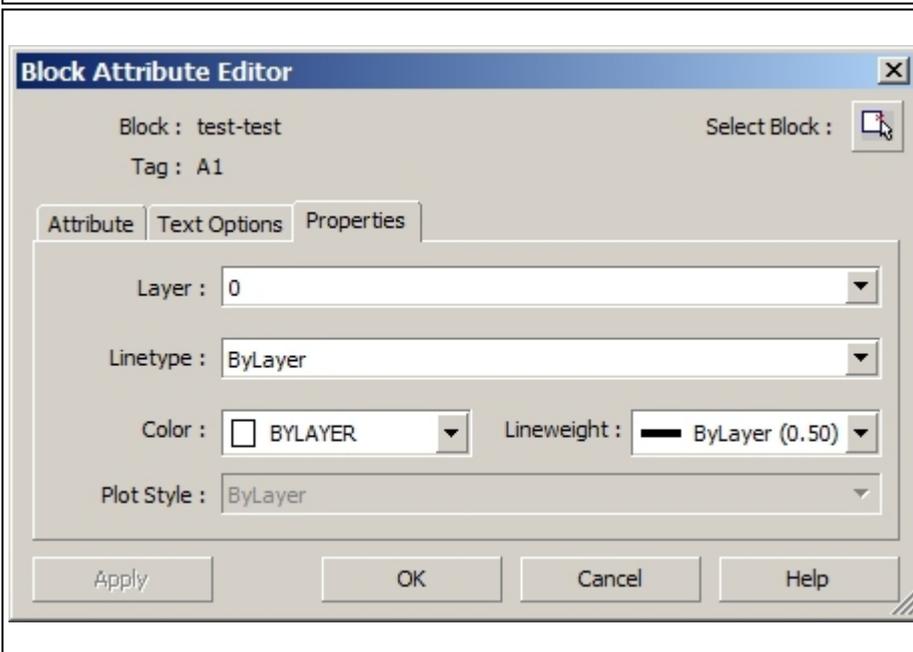
**Rotation** specifies the text rotation angle assigned to the attribute

**Backwards** specifies whether the attribute is displayed backwards

**Upside down** specifies whether the attribute is displayed upside-down

**Width Factor** specifies the text width assigned to the attribute

**Oblique Angle** specifies the text oblique angle assigned to the attribute



**Layer** specifies the layer assigned to the attribute

**Linetype** specifies the linetype assigned to the attribute

**Color** specifies the color assigned to the attribute

**Lineweight** specifies the lineweight assigned to the attribute

**Plot Style** specifies the plotstyle assigned to the attribute (disabled when the drawing is in color-dependent mode)

This dialog is free resizeable, and will maintain size and position of the dialog across Bricscad sessions.



Compile

## Compiles SHP shape files into SHX files

-  Command: **Compile**  
Opens the standard file selection dialog to select a ANSI shape file to be compiled into SHX format

The **Compile** command is able to compile both Symbol as well as TextFont shape files (ANSI + UNICODE) into SHX file format.

### Remarks :

Currently, UNICODE BigFont shp files can not be compiled into shx format !



## Dxbln

---

### Imports DXB (Drawing eXchange Binary) encoded binary files into the drawing

-  Command: **Dxbln**  
Opens the standard file selection dialog to select a \*.dxb file for import

### Remarks :

Please do not confuse - DXB format is not a binary DXF; instead, it is an old vector format to exchange vector graphics;  
it is (amongst other tools) created by AutoCAD DXB plotter/printer driver.



## LayerTools

---

### Dialog to manage drawing layers based on contained entities

-  Command: **LayerTools, ddITools**

The LayerTools command opens a small "modeless" (or "parallel") dialog, to provide a set of layer handling functions designed for fast and comfortable layer control, based on selected entities to determine target layers to be effected.



Compared to numerous similar layer management tools, the LayerTools provide a number of advanced functionalities like **Reverse Selection**, **Nested Selection**, and **Layer Restore**, as well as a comfortable user interface with some nice features like **Auto-Minimize** and **Mouse Tracking**.

Please see [General Hints](#) for some more details regarding LayerTools' special features.

All functions and features are documented in detail under [Using LayerTools](#).

In addition, one of the next BcadTools Freeware version will provide extra toolbars for LayerTools functions - to match personal taste as best as possible.



## General Hints

### General Hints on LayerTools features

As an introduction, this chapter describes some general features.

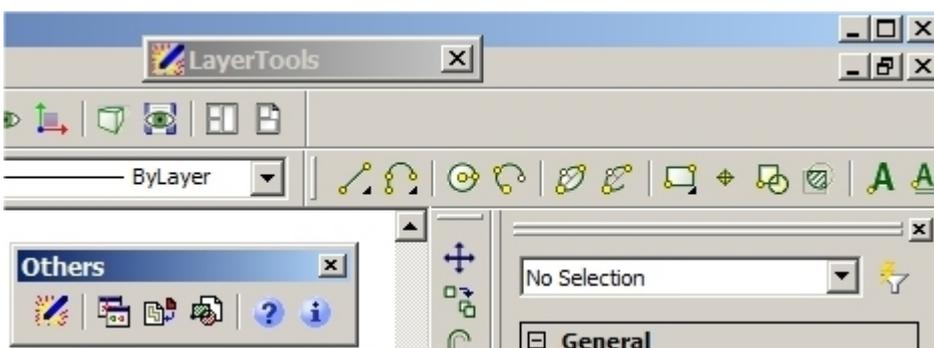
#### Automatic Dialog Collapse & Expand (Auto-Minimize)

Similar to a toolbar, the LayerTools dialog window always remains in foreground.

To cover as less drawing area as possible, the LayerTools window offers the **Auto-Minimize** feature :

whenever the mouse pointer leaves the LayerTools dialog and moves into Bricscad drawing area, the LayerTools dialog will automatically collapse to a small icon.

In opposite, if the mouse pointer moves into the LayerTools icon area again, the dialog expands to normal size again.



This behaviour does not only preserve effective drawing area, but also saves permanent mouse clicks ...

So LayerTools dialog can be placed somewhere, covering otherwise "dead space" - then, it does not even cost any productive screen area.

## Automatic Dialog Restart

The LayerTools dialog permanently maintains its position and status, even across Bricscad sessions. In other words, LayerTools window always (re-)opens in last screen position and status.

If LayerTools dialog was open when last Bricscad session was closed, it will automatically load and open LayerTools in next Bricscad session again; in opposite, if LayerTools dialog was closed, when Bricscad session finished, then LayerTools will not automatically load and open in next Bricscad session.

## Reverse Selection

Probably the most powerful and effective LayerTools feature is provided by Reverse Selection; this means, the layer operations like Off, Freeze, Lock, Unlock **will NOT process selected layers**, as picked by reference entities, but instead, these functions **will process NOT-selected layers**. In other words, those selected layers will be excluded from layer operation ...

This allows very elegant work flow : in example, to modify entities on particular layer(s) only, the **Lock** function with option **Not-selected** set active can be used to lock all other layer(s) and keep selected layer(s) unlocked.

Other example : the usual "Isolate" operation is provided as **Off** or **Freeze** using selected layers, with option **Not-selected** set active.

## Nested Layer & Entities Selection

LayerTools also provides the ability to process nested layers inside blocks or xrefs - this way it is easy to control layers which are otherwise somewhat more difficult to access.

To activate nested selection, simply answer with <return/space> as empty input, whenever the prompt message looks like :

\* **Select entity on target layer <sub-entity>** :

Here is an example, to change nested layers to OFF state :

\*\*\* OFF all selected entities' layers \*\*\*

\* Select entity on target layer <sub-entity> :

Select entities: <return/space>

\* Select Sub-entity on target layer <Exit> : <5>

\* Select Sub-entity on target layer <Exit> : <4>

\* Select Sub-entity on target layer <Exit> :

## Layer Restore

Another powerful feature is available with **Restore** function.

LayerTools records all layer state changes - using **Restore** function, layer state changes can be made undone stepwise and without limits, in modelspace, paperspace and layouts. All changes to drawing entities, and all changes to layer properties, are not effected ! This way, LayerTools allows comfortable entity editing and restoring layers to their previous states !

In example, if a number of layers have been set to Locked or Frozen or Off, to allow easier entity editing on remaining layers, it is very easy to go back to previous layer situation by Restore function - all layers are reset to their previous status.

There is no limit regarding the depths of save + restore operation for this kind of "Layer Undo"; Saving and Restoring layer status is effective for layer status On/Off, Thawed/Frozen,

Unlocked/Locked, and Current layer.  
Any other layer status and property is not effected.



## Using LayerTools

### Using LayerTools functions

LayerTools functions will process those layers of selected (or not-selected) entities. When a function starts, the user is asked for entities selection first - select the entities to determine the layers to be processed, or answer with <return/space> to switch to nested entity selection.

It is also possible to answer with <return/space> at the entities selection prompt: now it is possible to select sub-entities to specify nested layers (like layers of block or xref entities).

Here is an example, to change nested layers to OFF state :

```
*** OFF all selected entities' layers ***
* Select entity on target layer <sub-entity> :
Select entities: <return/space>
* Select Sub-entity on target layer <Exit> : <5>
```

	<p><b>All Layers</b> functions <b>Off</b>, <b>Thaw</b> and <b>Unlock</b> will set all drawing layers to Off, Thawed and Unlocked state</p> <p><b>Layer functions</b></p> <ul style="list-style-type: none"> <li>Current changes the current layer to reference entity's layer</li> <li>Freeze changes all reference entities' layer to frozen</li> <li>Off changes all reference entities' layer to off</li> <li>VP-Freeze changes all reference entities' layers to vp-frozen</li> <li>Lock changes all reference entities' layer to locked</li> <li>Unlock changes all reference entities' layer to unlocked</li> </ul> <p><b>Entities functions</b></p> <ul style="list-style-type: none"> <li>Copy copies all entities on reference entities' layers to another layer</li> <li>Move moves all entities on reference entities' layers to another layer</li> <li>Show QuickView all entities located on layers of reference entities</li> <li>Erase erases all entities located on layers of reference entities</li> <li>Foreground positions all entities located on layers of reference entities into foreground</li> <li>Background positions all entities located on layers of reference entities into background</li> </ul> <p><b>Restore</b> restores the previous layer situation</p>
--	--

### Option Copy/Move All

This option only effects the **Copy** and **Move** functions !

If active, Copy and Move will process all entities located on selected target layers - otherwise, if not

active, Copy and Move will process directly selected entities only.  
Depending on the status of this option, LayerTools adjusts the tooltips for Copy and Move buttons, to prevent confusions.

### Remarks :

The Restore stack is unlimited in size and depth, this allows to restore all previous layer situations even back to the beginning of your LayerTools operations.



## MLine

---

### Draws MultiLine entities



Command: **MLine**

Current settings: Justification = Top, Scale = 20.0000, STyle = STANDARD

Specify [start point](#) or [[Justification/Scale/STyle](#)]:

#### Start Point

Enter the start point of the multiline.

After the start point has been specified, the prompt message changes into :

Specify next point:

to specify the second point of the multiline.

If the multiline has 2 point vertex points, the prompt message changes into :

Specify next point or [[Undo](#)]:

If the multiline has more than 3 vertex points (in other words, at least 2 segments), the prompt message changes into :

Specify next point or [[Undo/Close](#)]:

#### Undo

Undoes the last vertex of the multiline, and displays a new prompt, depending of multiline vertex number.

#### Close

Joins the last multiline vertex with the first vertex to create a closed multiline.

#### Justification

Specify the "vertical" orientation of the multiline. A new prompt message is displayed :

Enter justification type [[Top/Zero/Bottom](#)] <Top>:

#### Top

Orients the multiline in a way that the line element with most positive offset is drawn between the vertex points.

## Zero

Orients the multiline in a way that the (real or imaginary) line element with offset 0.0 is drawn between the vertex points.

## Bottom

Orients the multiline in a way that the line element with most negative offset is drawn between the vertex points.

The offset value for each multiline element can be defined in [Multiline Style](#) dialog.

## Scale

Specify the multiline scale. A new prompt message is displayed :  
Enter MLine scale <20.0000>:

This scale factor is directly applied to offset values of each multiline element.  
A negative scale factor will "invert" or flip the sequence of multiline elements.

The offset value for each multiline element can be defined in [Multiline Style](#) dialog.

## Style

Specify the multiline style to be used for the multiline. A new prompt message is displayed :  
Enter MLine style name or [?]:

You can enter the name of an already loaded multiline style, or enter ? to get a list of all loaded multiline styles.

The multiline style will become the active style (system variable CMLSTYLE).

Multiline styles can be defined and loaded with [Multiline Style](#) command.



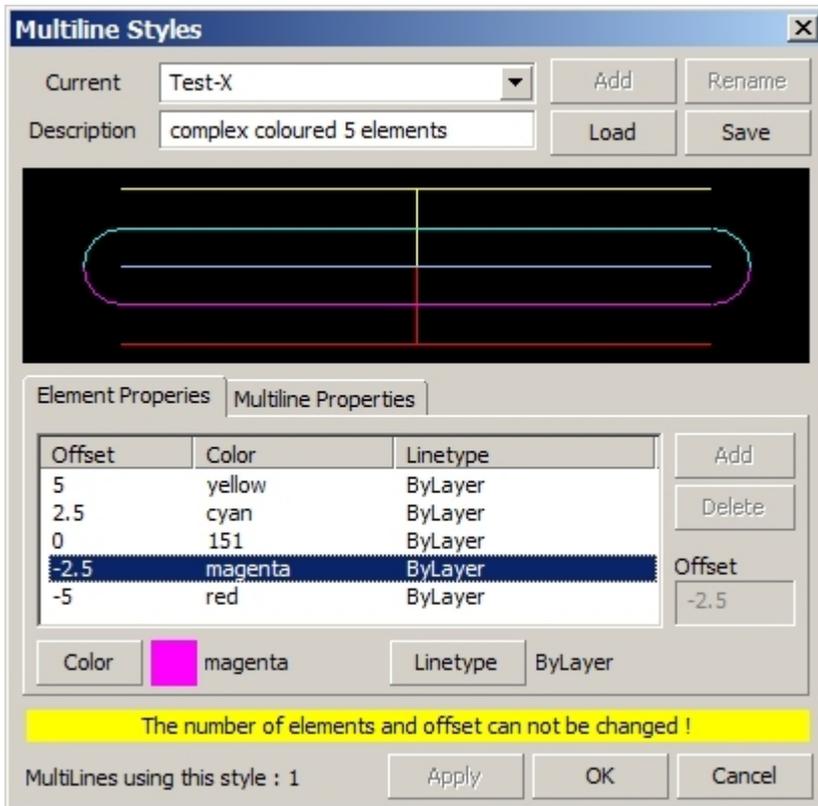
## MLStyle

---

### Create, Modify and Manage MultiLine styles

 Command: **MLStyle**

This command opens the MultiLine Style dialog.



The dialog always selects the drawing's current multiline style (system variable CMLSTYLE) in style popup list - so the properties of active multiline style are displayed initially.

If the drawing contains multiline objects using a selected style, you will see a yellow hint message at dialog bottom : because the style is referenced, the number and offsets of multiline style elements can not be changed.

Only for unreferenced multiline style it is possible to change the number and offsets of multiline elements.

The selected multiline style is always previewed - any changes made to the multiline style elements and properties are immediately shown in the preview image.

## Current

Shows the current Multiline style (CMLSTYLE); the popup listbox lists all multiline styles already loaded into the current drawing.

Selecting a different multiline style from the listbox will set this style as the current one (after using Apply or OK); the preview graphic will immediately show the graphics for selected multiline style.

You can also edit the multiline style name directly - this will activate the Add and Rename buttons, if the style name does not already exist.  
(the "Standard" style can not be renamed !).

## Add

Using the Add function you can create a new multiline style, using the actual style settings.

## Rename

This function allows to rename an existing multiline style to a new name (except for style "Standard").

## Load

Opens the "Load Multiline Styles" dialog to specify a multiline style definition file (\*.mln). After choosing a \*.mln file, the dialog lists all contained multiline styles, and allows to select a particular multiline style to be loaded into the drawing.

## Save

Saves the current multiline style as a multiline style definition (\*.mln) file.

## Apply

If there are changes to selected multiline style, the Apply function will save the changes to multiline style definition, and also update all effected multiline entities in the drawing.

To modify the properties of multiline elements (the parallel lines), please see [Edit Style Elements](#).  
To modify general multiline properties, please see [Edit Style Properties](#).



## Edit Style Elements

---

### Edit Properties of Multiline Style Elements

For each element of a multiline element, the offset (to 0.0), color and linetype can be specified. Any change to these properties is immediately updated in multiline style preview.

#### Add

Adds a new multiline element at specified offset.

#### Delete

Deletes the selected element from style (except for last element).

#### Offset

Defines the offset for selected or new multiline element (for selected element, press enter to update the list and preview)

Offset	Color	Linetype
1	BYLAYER	Continuous
0	50	ByLayer
-1	BYLAYER	Continuous

Buttons: Add, Delete, Offset (0)

Color: 50, Linetype: ByLayer

## Color

Opens the standard color selection dialog to choose the color for selected or new multiline element.

## Linetype

Opens the standard Linetype selection dialog to choose the linetype for selected or new multiline element.



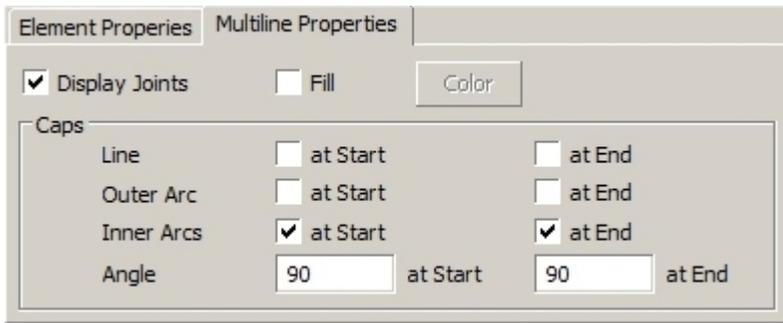
## Edit Style Properties

---

### Edit general Properties of Multiline Styles

For each element of a multiline style, a number of general properties are available to control the visual representation of a multiline.

Any change to these properties is immediately updated in multiline style preview.



### Display Joints

This option determines whether joint (or "miter") lines between the multiline elements at each vertex.

### Fill

Determines whether the multiline uses a background fill color.

### Color

If **Fill** is active, the color selection dialog opens to select the fill color

### Line at Start

This option determines, whether a line segment across multiline elements is shown at the start side of the multiline.

### Line at End

This option determines, whether a line segment across multiline elements is shown at the end side of the multiline.

### Outer Arc at Start

This option determines, whether an arc between the 2 outermost multiline elements is shown at the start side of the multiline.

### Outer Arc at End

This option determines, whether an arc between the 2 outermost multiline elements is shown at the end side of the multiline.

### Inner Arcs at Start

This option determines, whether arcs between inner pairs of multiline elements is shown at the start side of the multiline.

If the style has an odd number of elements, the central line elements remains unconnected.

### Inner Arcs at End

This option determines, whether arcs between inner pairs of multiline elements is shown at the end side of the multiline.

If the style has an odd number of elements, the central line elements remains unconnected.

### Angle at Start

Specifies the angle for the end cap at the start side of the multiline. This angle is meant as an offset to mathematical 0 degrees.

### Angle at End

Specifies the angle for the end cap at the end side of the multiline. This angle is meant as an offset to mathematical 0 degrees.

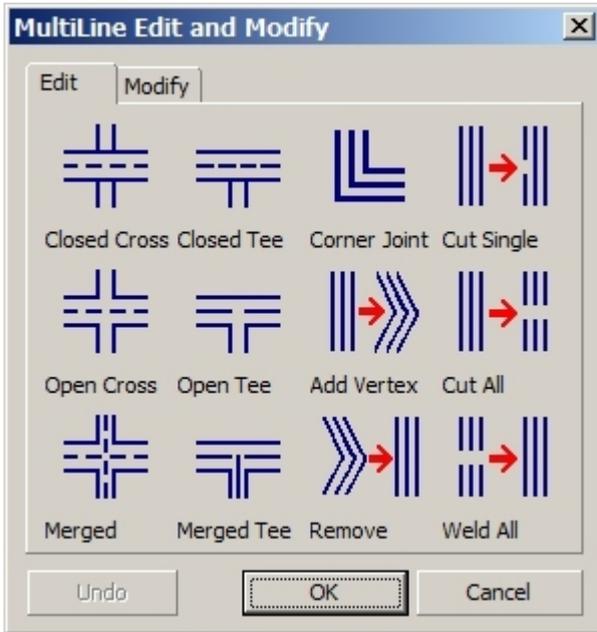


## MLEdit

### Edit MultiLine vertices, cuts / breaks, and intersections

 Command: **MLEdit**

This command opens the MultiLine Edit dialog.



There are several utilities provided to modify multiline crossings, connections and vertex sequence.

**Due to missing API support in DwgDirect and Bricscad, most of these functions are not yet available - but we are working on it to get some of the more important functions to be implemented in near future.**

**We are very sorry for this inconvenience !**



#### Closed Cross

creates a "closed cross" intersection from 2 selected multilines - the first selected multiline has priority.



#### Open Cross

creates an "open cross" intersection from 2 selected multilines - for the second multiline, only the outer-most lines will get breaks.



#### Merged

creates an "merged cross" intersection from 2 selected multilines - both multilines have same priority.



#### Closed Tee

creates a "closed tee" intersection - the first multiline is trimmed resp. extended to the second multiline.



#### Open Tee

creates an "open tee" intersection - the first multiline is trimmed resp. extended to the second multiline, which is also opened.



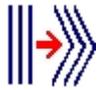
#### Merged Tee

creates a "merged tee" intersection - the first multiline is trimmed resp. extended to the second multiline.



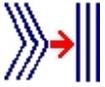
### Corner Joint

creates a "corner joint" intersection - the elements of both multilines are trimmed resp. extended to their intersections.



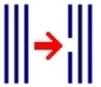
### Add Vertex

inserts a new vertex into the multiline.



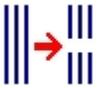
### Remove Vertex

removes a vertex from the multiline.



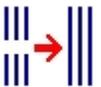
### Cut Single

inserts a cut into selected element of a multiline.



### Cut All

inserts a cut into all elements of a multiline.



### Weld All

removes all cuts from all multiline segments.



## MLModi

---

### Edit Properties of MultiLine

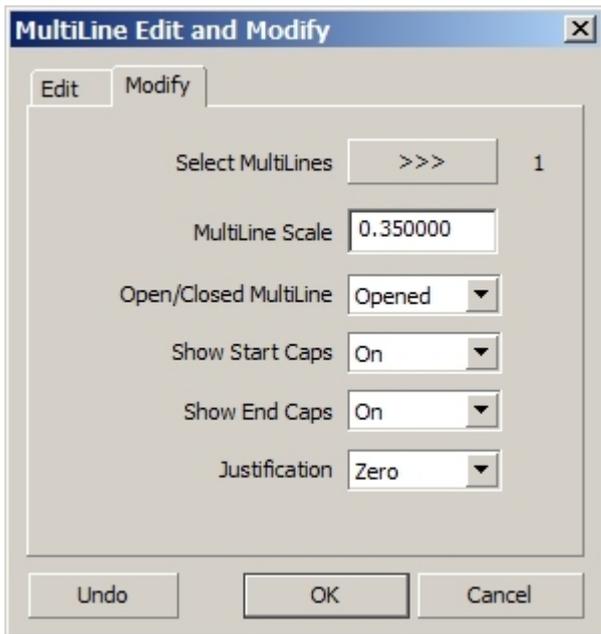


Command: **MLModi**

Select MultiLines: <select one or more multiline objects>

When no multiline is selected, the command finishes without opening the dialog.  
If at least 1 multiline object is selected, MultiLine Modify dialog is opened.

The dialog shows all multiline properties, that are stored "per entity" - these properties are taken from multiline style when the multiline is created, but are entity-specific afterwards.



If exactly 1 multiline is selected, all properties are directly shown;

if there are 2 or more multilines selected, particular properties will only be shown, if all multilines have the same property values - otherwise, the field keeps empty to indicate different values (similar to Bricscad's PropertyManager).

Any change of properties will immediately update all selected multilines (see [Undo](#)).

### Select MultiLines

At any time, you can select other multiline(s) for editing. Previously selected multilines will keep their current properties (no Cancel of previous edit).

### Undo

The dialog has a separate Undo stack, each property change can be undone, until all changes are undone - if no Undo data are available, the button is disabled.

### MultiLine Scale

Changes the scale factor for selected multiline(s) - press <enter> or <tab> key or click into any other listbox to finish editing and to update the multiline(s).

### Open / Closed MultiLine

Sets selected multiline(s) to opened or closed state.

### Show Start Caps

Sets selected multiline(s) to show caps graphic at start point - or not.

### Show End Caps

Sets selected multiline(s) to show caps graphic at end point - or not.

### Justification

Sets selected multiline(s) to Top, Zero or Bottom justification.



## Toolbars

### Available Toolbars

There are a number of toolbars provided with the menu :



Commands Mline, MIEdit, MIModi, MIStyle



Commands BattMan, BattEdit, AttSync, AttRedef



Commands LayerTools, Arx, Dxbln, Compile, Help, Info



# Programming Interfaces

---

## Programming Interfaces (API) provided for BcadTools

BcadTools Freeware provides a number of programming interfaces for its commands and functions. This allows other BRX and Lisp based applications to integrate BcadTools features.



# Translations

---

## Translating BcadTools Freeware

By default, BcadTools provides the user interface in English and German language - but the software is specifically designed to support an easy translation into any other language.

There is a kind of automatism in the way as BcadTools identifies which language dependent files are to be used :

- 1. step** If there is a user-defined language (stored in Registry), then those related language dependent files are used
- 2. step** if such files are not found in BcadTools installation folder, BcadTools tries to use use the same language key as Bricscad
- 3. step** if such files are not found, the default English files with <en\_US> language key are used
- 4. step** if such files are also not available, BcadTools will use files without language key, if present

There are 2 main subjects for translations - the [user interface](#) and this [documentation](#).

As specified in [License Agreement](#), there are no legal restrictions to create more language files for BcadTools. We would only like to ask to send translated files to us, to have more translations included in BcadTools by default.

Please send your translations to [bcadtools@LT-Extender.com](mailto:bcadtools@LT-Extender.com) - naturally, the translated files will get the copyright remark of its author !



## Translating User Interface

---

### Translating the user interface

All command messages and dialog texts are located in a single file - **BcadTools.<language>.msg**. Here, the <language> item within the filename refers to the localization version of Bricscad.

These files are normal ANSI files - please do not save as Unicode or UTF-8/16 format ! Each line in the file represents a single message or dialog label, using a simple

**<key> # <message/label>**

format; <key> is an internal identifier and must not be changed at all.

### Procedure for translation

lets assume the translation into French language - Bricscad uses <fr\_FR> language key here. Thus, the file **BcadTools.en\_US.msg** can be copied to **BcadTools.fr\_FR.msg**, which can be used for French translation then.

Then, any suitable text editor (supporting ANSI file save) can be used to edit the message and label texts.

If a French version of Bricscad is running, BcadTools will automatically use the new

**BcadTools.fr\_FR.msg** message file.

**Remarks :**

all message files must be located in **.Language** sub-folder in BcadTools Freeware installation folder !



## Translating the Menu

---

### Translating the Menu

All command messages and dialog texts are located in a single file - **BcadTools.<language>.cui** and **BcadTools.<language>.mns**.

Here, the <language> item within the filename refers to the localization version of Bricscad.

The old-style mns files are provided to make translation easier - so it is suggested to translate the mns file, and then to convert into cui file using Bricscad's `_MENULOAD` command by loading the mns file.

### Procedure for translation

lets assume the translation into French language - Bricscad uses <fr\_FR> language key here.

3. copy the file **BcadTools.en\_US.mns** to **BcadTools.fr\_FR.mns**, which can be used for French translation then
4. copy the file **BcadTools.en\_US.dll** to **BcadTools.fr\_FR.dll**, which contains the menu icons only (Resource-DLL)

Then, any suitable text editor (supporting ANSI file save) can be used to edit the menu entries.

It is strongly suggested to keep the English command names (including the "\_" prefixes !) for the menu items - this will ensure the menu to work properly, even if the [message file](#) is not yet or correctly translated.

If a French version of Bricscad is running, BcadTools will automatically use the new **BcadTools.fr\_FR.mns** resp. **BcadTools.fr\_FR.cui** menu file.

**Remarks :**

all menu files must be located in **.Language** sub-folder in BcadTools Freeware installation folder !



## Translating Documentation

---

### Translating the documentation

All the 3 help and documentation files (\*.chm, \*.htm, \*.pdf) are created from a single file in special **HelpNDoc** format (see [www.helpndoc.com](http://www.helpndoc.com)). There is a free version, which allows to create and edit the documentation in comfortable way.

For the CHM, HTM and PDF based documentation files, the same logical rules as for message files apply - a French version of CHM, HTM and PDF files would be named BcadTools.fr\_FR.chm, BcadTools.fr\_FR.htm and BcadTools.fr\_FR.pdf.

At the moment, we only provide documentation files in English version - thus, the <en\_US" language key used. This will be extended, once the German and other versions of help and documentation files are provided.

For parties interested in translation of the documentation files, we will provide that **HelpNDoc** file - please send us an email to [bcadtools@LT-Extender.com](mailto:bcadtools@LT-Extender.com).

**Remarks :**

all documentation files files must be located in **.Help** sub-folder in BcadTools Freeware installation folder !



## Feedback

---

### **Send us Feedback !**

For any problem you encounter, or for any idea, wish or hint, please send us an email to

[bcadtools@LT-Extender.com](mailto:bcadtools@LT-Extender.com)

We will do our best to fix all problems, and to integrate customer ideas and wishes - many thanks in advance.

