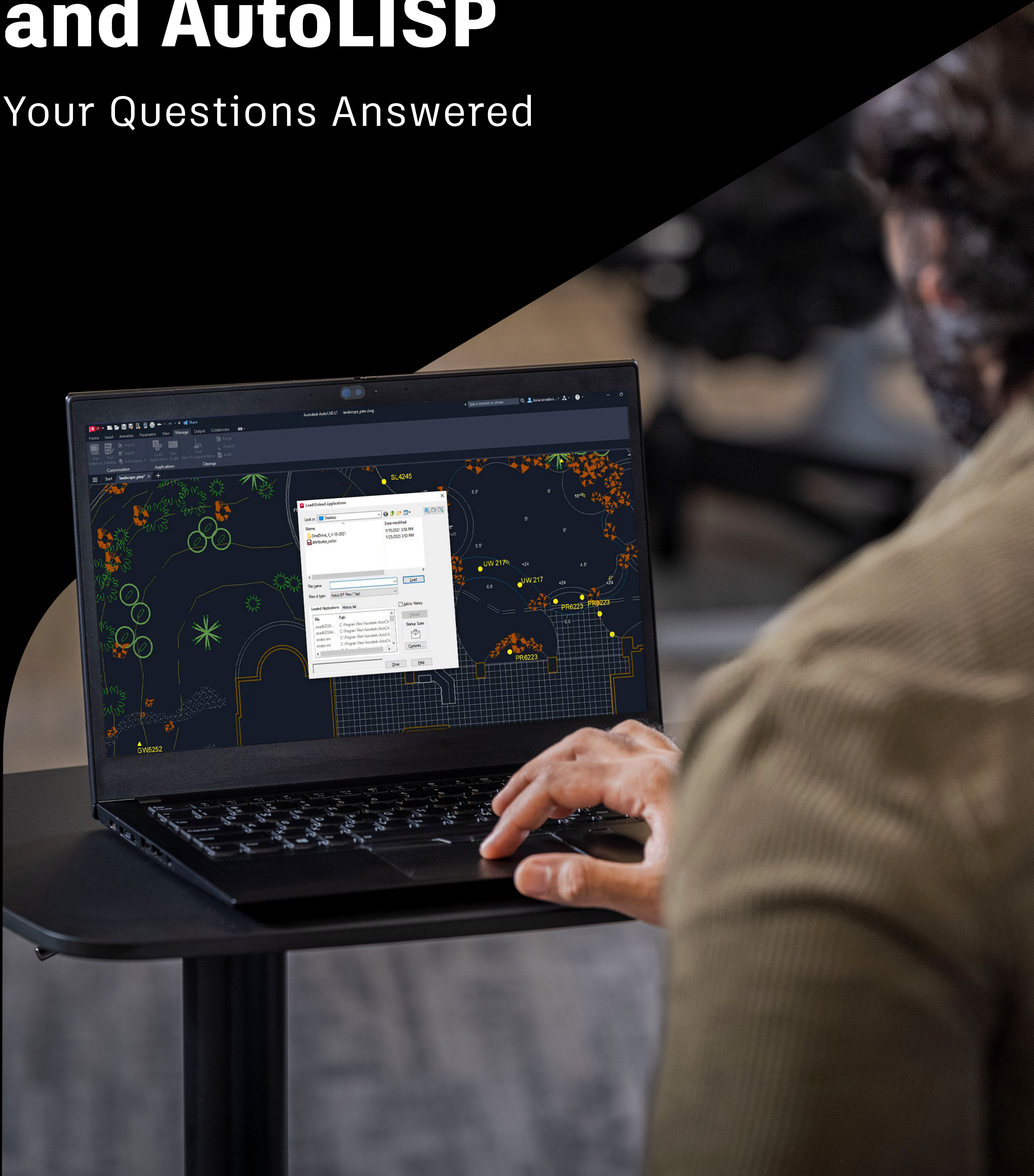
AUTODESK

How to Use AutoCAD LT and AutoLISP

Your Questions Answered



Introducing AutoLISP in AutoCAD LT

AutoCAD LT customers received a lot of value with the release of AutoCAD LT 2024. For the first time, AutoCAD LT 2024 now includes AutoLISP—further enabling organizations to streamline workflows, enforce CAD standards, and leverage thousands of pre-existing automations for AutoCAD-based programs.

This is great news for the AutoCAD LT community. Not only does the existing base of AutoCAD LT users have a new tool to explore, but it opens an entire group of AutoCAD users to take advantage of AutoLISP. And you don't have to become a programmer to do it.

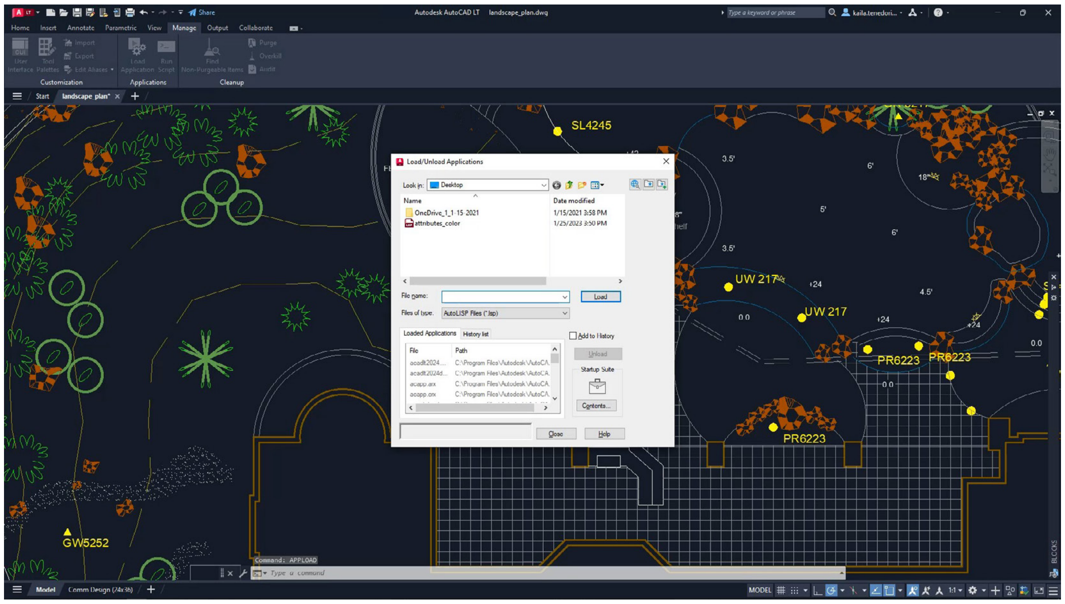
It's probably safe to say that most existing AutoCAD LT users don't know a lot about AutoLISP. And why would you? You couldn't use it, and nobody learns a feature or programming language they can't use. But there may be some of you who are aware (to some degree) of what AutoLISP can do. For both groups, there may be quite a few questions. Now that you have AutoLISP, what can you do with it? Where can you go to find code? What do you do with the code once you have it?

Not to worry! Here, we'll explore what you now have and how to find and use AutoLISP code that you can trust and help you in your daily tasks.

AutoLISP 101

AutoLISP is a dialect of the Lisp programming language. Developed in 1958, Lisp is the second oldest language still in use, next to Fortran. AutoLISP was added to AutoCAD version 2.18 back in 1986 as an interface to extend its functionality.

Not only can you use it to automate tasks similar to (but much more powerful than) scripts, but it also lets you interact with the editor in familiar ways. AutoLISP lets you specify points, select objects, enter alpha-numeric data at the command line, or provide input via dialog boxes that are created with its companion language, DCL (Dialog Control Language).



```
Command: APPLOAD
```

Quick Tips to Begin Using AutoLISP

One of the most important differentiators between AutoCAD and AutoCAD LT is the fact that AutoCAD LT users can now use AutoLISP code. VLIDE, which stands for Visual LISP Integrated Development Environment, is not included in AutoCAD LT. It's integrated into full AutoCAD and is where AutoLISP code is typically written and debugged.

With or without the built-in LISP editor, AutoLISP is easy for non-programmers to use. One of the reasons it's easier is the fact that AutoCAD LT is its own runtime interpreter for your code. In other words, you can enter a piece of AutoLISP code right at the Command prompt, and it will run it for you.

How can this be of value to you? Well, AutoLISP can do math. Maybe you'll find it quicker to use a bit of code instead of using the QuickCalc tool or even the Windows calculator. The syntax may be a little weird for you, as the operator or function comes first, so to add 2 and 2, you'd enter `(+ 2 2)`. It will return a value of 4. Please remember that an AutoLISP statement always returns a value. That means you can use an AutoLISP statement to respond to an AutoCAD prompt.

Let's say AutoCAD LT is asking for a distance, and you need to calculate the value first. We'll say it's Pi times 12 divided by 2 (because, why not?). Enter `(/ (* pi 12) 2)` at a distance prompt. AutoLISP returns 18.8496 as the distance (depending on your units setting).

What if you want the value as the string on a multileader?
Enter `(rtos (/ (* pi 12) 2)  4 2)` when prompted for text. That rtos function converts the value into a string or, in this case, 1’ - 6 3⁄4”.

Maybe you have a complex set of objects to select, and you know you will need to select them a few more times in your session. No problem when you have AutoLISP by your side. Just type in `(setq SS (ssget))`. The ssget function prompts you to select objects like you’re used to doing. Now go ahead and perform your complex selection.

Basically, the code you entered (the setq) will assign the selected set of objects to the variable SS, which AutoLISP will remember while the drawing remains open. It also gives you an easy mechanism for retrieving it. So, the next time you need these objects, enter `!SS` at the Command prompt. Just like that, you have the exact same selection set. As you find these little nuggets, copy and paste them into a text document that you keep at easy reach. Just edit a little text if you need to, then copy and paste!

One thing to note about AutoLISP code. Always use plain text, never anything formatted. Notepad is perfect for this. Just stay away from your favorite word-processing program.

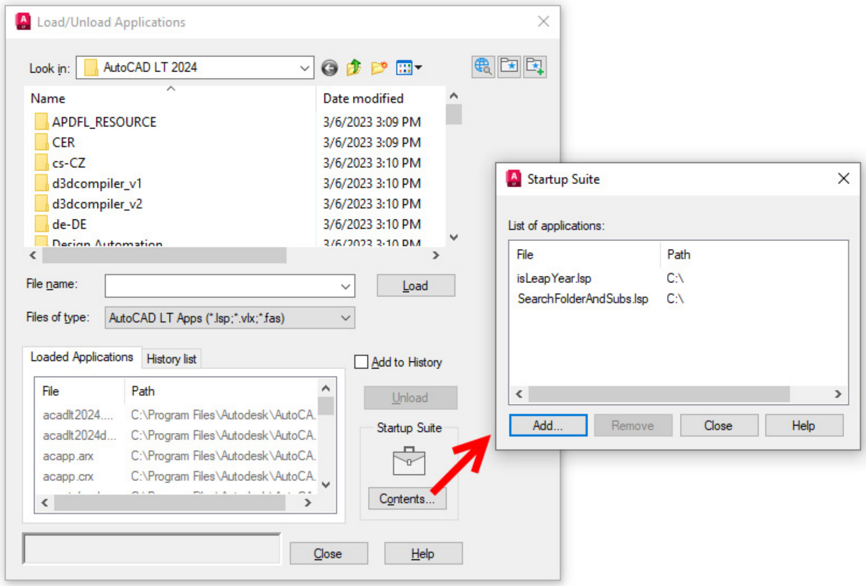
Loading and Running AutoLISP Code

OK, code snippets at the Command prompt are nice, but you’ll probably use your favorite search engine and find full-fledged AutoLISP routines on the Internet, which will be even more helpful. They’ll usually be in a single file with the extension of .lsp, so you’ll need to know how to load them and what to do to make them run.

Note: AutoCAD LT also allows you to load and run .vlx and .fas files, which are both a compiled format, so you won't be able to open and read them.

The easiest is the APPLOAD command. You can also find its icon on the Applications panel of the ribbon's Manage tab. Running it will display the Load/Unload Applications dialog, which is where you'll navigate to your AutoLISP file and click load. Pretty easy. But doing it that way means you have to do it every time you re-enter your drawing. AutoLISP files are not persistently loaded by default. In the lower right corner of the Load/Unload Applications dialog is something called the Startup Suite. If you put your file here, it will load whenever you open a drawing.

You can also use AutoLISP itself to open your file. Just type in `(load "filename.lsp")`, and you're good. That is if the file is in a defined search path. If not, you'll have to supply the full path before the file name. And that requires some AutoLISP knowledge, so you may not want to use this method.



Name	Date modified
APDFL_RESOURCE	3/6/2023 3:09 PM
CER	3/6/2023 3:09 PM
cs-CZ	3/6/2023 3:10 PM
d3dcompiler_v1	3/6/2023 3:10 PM
d3dcompiler_v2	3/6/2023 3:10 PM
de-DE	3/6/2023 3:10 PM
Design Automation	3/6/2023 3:10 PM

File	Path
acadlt2024....	C:\Program Files\Autodesk\AutoCA
acadlt2024d...	C:\Program Files\Autodesk\AutoCA
acapp.arx	C:\Program Files\Autodesk\AutoCA
acapp.crx	C:\Program Files\Autodesk\AutoCA

File	Path
isLeapYear.lsp	C:\
SearchFolderAndSubs.lsp	C:\

Here's another great tip. You can also drag and drop an AutoLISP file into your LT session. It will load, and you'll be ready to go. Or will you? Your AutoLISP file is now loaded, but how do you make it run? Sometimes the programmer is friendly and will include a prompt that tells you what to type in, so keep an eye on the Command Line history. If not, you'll have to open the AutoLISP file (a double click should do the trick) and find the line that defines how to run the routine. Look for the following in the code:

```
(defun C:commandNameHere …
```

Capitalization doesn't matter, so don't worry about that. You're looking for the defun (which is used to define a function in AutoLISP) and the C:, which is what tells AutoLISP that the word following it is the command name. That's what you'll type at the Command prompt to run your AutoLISP routine.

AutoLISP to Automate

Code snippets are great, as is loading and running code. But there's more! How about the ability to run the code you want every time you open a file? Think about those implications.

You could analyze your layers to make sure they match your standards and fix them if they don't. Do the same with all your styles. You can check the file's name, folder, or path to ensure compliance based on the drawing type or discipline it belongs to. It's all easily done with code you can find on the internet with a few search terms typed into your favorite search engine.

How do you get started? There's a special file that will load every time a drawing is opened. In full AutoCAD, it's called ACADDOC.lsp, but in LT now, it's called ACADLTDOC.lsp. It doesn't exist to start with, so you'll have to create it (remember your friend, Notepad) and make sure it is in a folder that is in your search paths.

It will work like the Startup Suite in the Load/Unload Applications dialog, but instead of just loading the routines so that you have access to them, you can also run the commands.

Typically, this is also where you'll control your LT environment. There may be some system variables, or environment variables that you want to ensure are always set to your liking.

Here are examples of AutoLISP statements that a company might add to the ACADLTDOC.lsp file:

```
(setvar "FILEDIA" 1)

(setvar "CMDDIA" 1)

(setvar "ATTDIA" 1)

(setenv "ShowFullPathInTitle" "1")
```

They're just AutoLISP statements, so when the file loads, they execute automatically without needing to enter a command name. It's a terrific way to standardize your setup.

You can also put shortened or little helper commands in there. Just make sure you remember the command name (the part after the C:). Again, this code loads and is ready for you to use. Here is an example of a command with a shortened name similar to a command alias. Just type in MP to run MATCHPROP.

```
(defun c:mp () (command "_matchprop") (princ))
```

Or it may be some kind of helper command. The following code will flip your selected objects 180° around the point you define.

```
(defun C:flip ()  (command "_rotate" (car (entsel)) "" (getpoint) "180") (princ))
```

Exploring Visual Lisp

Everything we've talked about so far involves using raw AutoLISP. But you may run across an extension that's widely used. When looking at the code, keep your eye out for the following statement at the top of the file: (vl-load-com). That's your clue that the code may utilize ActiveX related functions. Also, look for commands that start with "vla-" or 'vlax-".

Finding this isn't necessarily a bad thing, as your new and improved LT does support its use. But depending on what it's trying to do, it may not work properly. For instance, ActiveX in AutoLISP can create and modify 3D objects, but since LT doesn't support 3D, it wouldn't work. The same would go for constraints, profiles, external applications, etc.

But here is one kind of ActiveX in AutoLISP code that you might find useful, and that should work for you. Wouldn't it be cool if something was lurking in the background of your drawing session, just waiting for one thing in particular to happen?

Think about this scenario. Exploding dimensions is strictly forbidden in your office. But it happens anyway. How great would it be to have a piece of hidden code that's programmed only to do something if the EXPLODE command was used on a dimension? The user tries it, and the code takes over and prevents it.

This is called event-driven programming and is accomplished using reactors. It's a bit advanced, but you can find working examples. The reactor and the function it runs when activated (the "callback function") could reside in the ACADLTDOC.lsp as well.

Finding AutoLISP and Visual Lisp Code

You might not initially write code from scratch, but you're going to find it mainly online. Oh, you'll (probably) eventually do what almost every AutoLISP programmer did to begin with. You'll find some code, try to read it, change something, and see if it still works.

Here's a little AutoLISP function that changes all text styles to use ROMANS.

```
(defun C:MakeRomans (/ rewind next name)

  (setq rewind T)

  (while (setq next (tblnext "STYLE" rewind))

    (setq name (cdr (assoc 2 next)))

    (if (not (wcmatch name "*|*"))

    (command "-style" name "romans" "" "" "" "" "" ""))

    (setq rewind nil))

  (princ))
```

It probably looks foreign to you, but looking closer, you may recognize some things in there. You know the STYLE command, and, of course, ROMANS is familiar. Actually, that entire line looks a lot like how you'd write it in a script. Maybe you're ready to see what happens if you change something.

Maybe the first thing you'd try is a different font. You change ROMANS to ARIAL. Save, load, run… and voila! It works!

Back to the Internet. That's how you're going to find your code initially. One thing I can almost completely guarantee you is that if you have a good idea for a task for AutoLISP to handle, you're not the first to do so. Another thing about AutoLISP programmers—they love to show off their work. It also means that they love to help one another. That's a good combination for you.

First, you'll need to become adept at using specific keywords and phrasing to find precisely what you need. Make sure to include the whole word of AutoLISP (not just Lisp). I always lead with AutoCAD as well. Don't include LT. There's no LT AutoLISP code out there—yet!

Use the verb or verbs that describe what your desired code will do, and use AutoCAD language. For instance, say Copy, not Duplicate. Leave out article adjectives and prepositions. Try to build a search query similar to this: AutoCAD AutoLISP Export Layer Table

In time, you'll find what does and doesn't work well for you. You'll also start to see a pattern in the results. Certain websites will appear often. You'll also get a feel for the ones where you get the best results. And vice-versa, of course.

You can even try some of the latest in artificial intelligence. You've undoubtedly heard about the AI chat sites that have popped up this year. Just ask one of them to write you some code! But proceed with caution as it provided results with some "rookie" mistakes. As quickly as technology is changing and evolving, it might be perfected by the time you read this.

Resources to Learn More

Hopefully, all of this has made you curious to learn more. And there are so many resources available to get started.

The best place to find help with understanding what your code means is Autodesk's own AutoLISP Developer's Guide. And if you're ready to go that far, maybe you'd like to go a little farther. There's even a Getting Started tutorial to help you on your way. And, of course, there are the AutoLISP forums. Online classes are available from Autodesk University as well.

With the release of AutoCAD LT 2024, you can also check out What's New or Changed with AutoLISP (AutoLISP) and improvements with the Functions By Functionality References in the AutoLISP Reference Guide.

Give AutoLISP in AutoCAD LT a Try

With access to AutoLISP in AutoCAD LT, there's a whole world of automation and time-saving code for you to explore.

If you're not already an AutoCAD LT subscriber, you can go here to download a free 30-day trial. Experiment a little, learn a little, and don't be afraid to give it a try.