

CB-LISPLOAD

Befehl	CB-LISPLOAD
BricsCAD	Lite, Pro, BIM, Mechanical und Ultimate
Zusatzmodul	nicht erforderlich
Stand	04.09.2026

Alle eigenen Routinen in einem Zug austauschen

CB-LISPLOAD meldet sämtliche Befehle **CB- ...** aus der laufenden Sitzung ab und lädt alle Routinen aus dem eigenen Ordner neu. Der Bestand wird dabei vollständig ersetzt, nicht ergänzt. Anschließend sorgt die Routine dafür, dass diese Fassung bei jedem Start geladen wird, und zeigt eine Liste der Befehle.

Ohne die Routine kennt BricsCAD einen geänderten Stand erst nach einem Neustart. Der Weg über **APpload** bleibt, verlangt aber je Datei einen eigenen Durchgang durch den Dialog.

Voraussetzungen

Alle Routinen liegen in einem gemeinsamen Ordner. Ob sie darin lose nebeneinander liegen oder jede in einem eigenen Unterordner, spielt keine Rolle. Der Ordner darf heißen, wie er will, und irgendwo liegen.

Einrichten

Einmal je Rechner:

1. Einen Ordner für die Routinen anlegen und alle Dateien hineinlegen.
2. **APpload** starten und allein CB-LISPLOAD.lsp hinzufügen und laden.



Das war alles. Die Routine erkennt an ihrem eigenen Speicherort, wo die übrigen liegen, richtet das automatische Laden selbst ein und zeigt die Befehlsliste. Es ist kein Pfad einzutragen und keine Systemvariable zu setzen.

Nach einem Versionswechsel wird Schritt 2 wiederholt, weil der neue BricsCAD-Ordner die Einrichtung noch nicht kennt.

Aufruf

1. `CB-LISPLOAD` eingeben.

Es gibt keine Rückfrage und keinen Dialog.

Gebraucht wird der Befehl, wenn sich während der Arbeit etwas geändert hat: nach dem Bearbeiten einer Routine, nach einer Umbenennung oder wenn eine neue Datei dazugekommen ist. Beim Programmstart läuft das Laden von allein.

Ergebnis

Alle Befehle aus dem Ordner stehen sofort zur Verfügung, und zwar in der Fassung, die gerade auf der Festplatte liegt. Die Zeichnung wird nicht verändert.

Die Befehlszeile meldet den Ablauf.

```
Ordner: /home/nina/Bricsys/BricsCAD/CUSTOM LISP
9 Befehle abgemeldet.
9 Dateien geladen, 10 Befehle.
Wird ab dem naechsten Start automatisch geladen.
```

Eine Datei mit einem Fehler wird benannt und mitgezählt.

```
FEHLER: /home/nina/Bricsys/BricsCAD/CUSTOM LISP/CB-TEST/CB-TEST.lsp
8 Dateien geladen, 1 mit Fehler, 9 Befehle.
```

Die Befehlsliste

Zum Schluss öffnet sich eine Textdatei mit allen verfügbaren Befehlen und ihrer Aufgabe. Sie liegt im temporären Ordner und heißt nach dem Tag ihrer Entste-



hung, zum Beispiel CB-Befehle_2026-09-04.txt. Gespeichert werden muss sie nicht. Wer sie behalten will, sichert sie aus dem Editor heraus weg.

Die Liste erscheint bei jedem Aufruf des Befehls und außerdem einmal beim Einrichten, also beim ersten Start und wieder nach einem Umzug des Ordners. Beim gewöhnlichen Programmstart erscheint sie nicht.

Die Aufgabe je Befehl stammt aus einer Zeile im Kopf der jeweiligen Routine:

```
;;; Zweck: Kopie auf dem aktuellen Layer
```

Fehlt die Zeile, bleibt die Spalte leer.

Festlegungen

Punkt	Verhalten
Suchbereich	der Ordner und seine direkten Unterordner, nicht tiefer
Abgemeldet werden	alle Befehle, deren Name mit CB beginnt
Ausnahme	CB-LISPLOAD selbst bleibt bestehen
Automatisches Laden	über on_doc_load.lsp im Support-Ordner
Beim Programmstart	nachgeladen wird nur, was noch nicht geladen ist
Befehlsliste	temporärer Ordner, Name mit Datum, wird nicht aufgehoben

Wie der Ordner gefunden wird

Die Routine fragt BricsCAD nach den geladenen LISP-Dateien und sucht darin ihre eigene. Aus deren Pfad ergibt sich der Ordner.

Liegt CB-LISPLOAD.lsp in einem eigenen Unterordner mit diesem Namen, gilt der Ordner darüber. Liegt sie lose zwischen den anderen Dateien, gilt ihr eigener Ordner. Beide Ablagen kommen vor, und beide funktionieren ohne Einstellung.

Wie das automatische Laden eingerichtet wird

BricsCAD führt die Datei on_doc_load.lsp im Support-Ordner bei jeder Zeichnung aus. Die Routine schreibt dort einen Block, der sie selbst lädt:



```
;;; --- CB-LISPLOAD ---  
(if (findfile ".../CB-LISPLOAD.lsp")  
    (load ".../CB-LISPLOAD.lsp"))  
;;; --- CB-LISPLOAD ---
```

Geladen wird also nur diese eine Datei, und die zieht beim Laden ihre Nachbarn nach. Stehen in on_doc_load.lsp bereits eigene Zeilen, bleiben sie unberührt. Nur der Block zwischen den beiden Markierungen wird ersetzt.

Beim Start prüft die Routine nicht nur, ob ein Block da ist, sondern ob er auf die Datei zeigt, die gerade läuft. Wird der Routinenordner verschoben, zeigt der alte Block ins Leere. Dann gilt die Einrichtung als nicht erledigt: Die Routine schreibt den Block neu und zeigt die Begrüßung wie beim ersten Mal.

Der Applikationsmanager wird dafür nicht benutzt. Sein Haken AutoLaden liegt an einer Stelle, die BricsCAD selbst verwaltet und die sich aus einer Routine heraus nicht zuverlässig setzen lässt.

Warum der ganze Bestand ausgetauscht wird

Beim Aufbau einer Sammlung ändern sich Namen. Wird eine Routine umbenannt, bleibt der alte Befehl in der Sitzung aufrufbar und führt alten Code aus. Das fällt erst auf, wenn jemand den alten Namen benutzt. Ein vollständiger Austausch macht die Suche nach dem Unterschied überflüssig.

Grenzen

- Nur eine Ordner Ebene tief. Ein Unterordner im Unterordner wird übersehen.
- Es wird geladen, nicht entladen. Eine Funktion, die kein Befehl ist, bleibt in der Sitzung bestehen, bis BricsCAD neu startet. Abgemeldet werden nur die Befehle.
- Enthält eine Datei einen Fehler, wird sie gemeldet und übersprungen. Die übrigen werden weiter geladen.
- Gefunden werden nur Dateien mit der kleingeschriebenen Endung .lsp. Unter Linux unterscheidet die Suche Groß- und Kleinschreibung.



- Im Applikationsmanager erscheinen die Routinen nicht. Er zeigt nur, was er selbst geladen hat. Was wirklich verfügbar ist, zeigen die Befehlsliste und die Eingabe von cb in der Befehlszeile.
- Die Befehlsliste enthält keine Umlaute. BricsCAD schreibt Textdateien als Windows-1252, und die Routine bleibt deshalb bei ae, oe und ue.
- Läuft die Routine, ohne über `APpload` geladen worden zu sein, findet sie ihren Ordner nicht und meldet das.

Stand

Befehl	CB-LISPLOAD
Datei	CB-LISPLOAD.lsp
Erstellt	25.05.2026
Geprüft	04.09.2026, BricsCAD V26 Lite unter Linux

