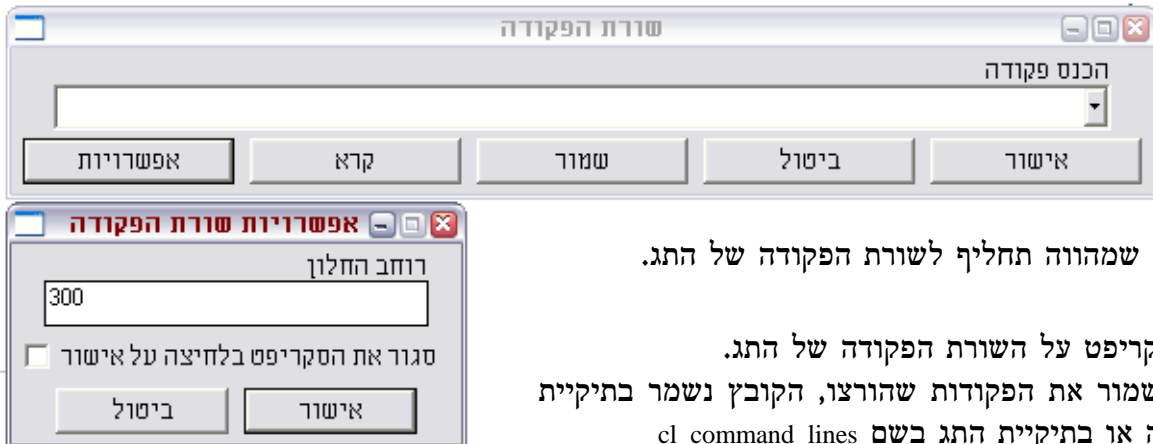


## שורת הפקודה



זהו סקריפט שמהווה תחליף לשורת הפקודה של התג.

יתרונות הסקריפט על השורת הפקודה של התג.

1. ניתן לשמור את הפקודות שהורצו, הקובץ נשמר בתיקיית העבודה או בתיקיית התג בשם `cl_command_lines` צריך ללחוץ קרא מקובץ. כיון שהסקריפט לא קורא מעצמו את הקובץ הזה. קריאת הקובץ משכתבת את רשימת הפקודות שנמצאות בזיכרון, והיא מוחלפת ברשימה שבקובץ. ניתן לראות קובץ זה בזיכרון התג ע"י קונטרול+K ברשימת המשתנים, ולערוך אותו אם רוצים.
2. ניתן למקם את החלון של הסקריפט היכן שרוצים. המיקום נשמר בקובץ `script.ini` שבתג, או בתיקייה הנוכחית אם כבר קיים כזה קובץ. גם אם לוחצים על ביטול, הסקריפט זוכר את מקומו.
3. ניתן לכתוב כמה פקודות בשורה עם המפריד `\r` או `\n` בין הפקודות, וכולם יבוצעו אחת אחרי השניה.
4. השורה ארוכה כדי לראות גם פקודות ארוכות.
5. מינימום רוחב החלון 100 ומקסימום 800.
6. כאשר מקלידים פקודה שכבר היתה ברשימה, הפקודה הקודמת נמחקת.
7. אם רוצים להריץ ע"י סקריפט אחר את הפקודה האחרונה שבוצעה הפקודה היא כזאת.  
`run file "zcl.lib" func "last_command"`
8. אני ממליץ להגדיר קיצור דרך לסקריפט זה את המקש ESC ע"י הסקריפט `zkb`. (ולמי שאין אז מקש F12).
9. הקשת אנטר מבצעת את הפקודה ומשאירה את החלון פתוח. קונטרול+אנטר מבצעת את הפקודה וסוגרת את החלון. (באפשרויות ניתן להגדיר שגם בהקשת אנטר החלון יסגר). הקשת ESC סוגרת את החלון.
10. ניתן לכתוב כל תו אסקי בצורה הזאת: #224 בשביל אות א #321 בשביל רווח. כובע ^ משמש כתו רווח.
11. ניתן להגדיר בקובץ `zcl.ini` תחת מילת קטע `[alias]` קיצורי פקודות.  
לדוגמא `a=msg "a" \r msg "b"` כאן מוגדר קיצור למקש `a`, כאשר נקיש `a` ואנטר יבוצעו שתי הפקודות `msg`. ניתן גם להגדיר פרמטרים. כאשר `%1` מייצג את הפרמטר הראשון `%2` את השני וכן הלאה. פרמטר נקרא מילה ללא רווחים, לכן אם רוצים להכניס פרמטר שיש בו תו רווח צריך להשתמש בתו כובע ^ או `#32`.  
לדוגמא, אם נגדיר `bb=msg "%2%1"` אז כשנקיש `bb a b` ואנטר, נקבל במסך הודעה עם `ba`  
דוגמא נוספת, אם נגדיר `bb=msg "%1"` ואז אם נקיש `bb a#32b` ואנטר, נקבל במסך הודעה עם `a b`  
בקובץ `ini` צריך להקפיד שלא יהיה מספר צמוד למספר של הפרמטר, (לדוגמא `%39` יובן כפרמטר ה-39 ולא כפרמטר 3 עם ספרה 9 לידו). הסקריפט תומך עד 99 פרמטרים.
12. ניתן לכתוב סקריפט עם משתנים ופונקציות בקובץ `ini` הזה כמו שתראה להלן בדוגמא המצורפת.  
על פי הדוגמא המצורפת ניתן להקיש פקודות כאלו  
כתיבת הפקודה `@ 51` ואנטר תמצא את הקוד 51.  
כתיבת הפקודה `:/` ואנטר תמצא נקודותיים  
כתיבת הפקודה `* / ?` ואנטר תמצא תו כל שהוא  
כתיבת הפקודה `F:` ואנטר תריץ סקריפט בשם `F.PRГ`.
13. אין הגבלה כמה קיצורי פקודות אפשר להגדיר.
14. מילת מפתח אינה רגישה לאותיות גדולות או קטנות באנגלית, לכן מילת מפתח `A` שווה למילת מפתח `a`

15. שים לב, הגדרת קיצור כל שהוא, מחייבת להכניס רווח בשורת הפקודה לפני כתיבת הפרמטר הראשון. אפשר לוותר על הרווח הזה, ובמקום לכתוב @ 51 לכתוב @51 והפקודה תעבוד. כדי שהסקריפט ידע לעשות זאת, צריך להוסיף בקובץ ini בקטע [options] מילת מפתח words את המילים שרוצים שיעבדו גם ללא רווח בין המילת קיצור לפרמטר הראשון. ראה בדוגמא בהמשך.
- שים לב! עלולה להיווצר התנגשות, אם הגדרנו קיצור אחד : וקיצור אחר :: הפיתרון לכך להקפיד שברשימת מילות המפתח words תהיה קודם המילה הארוכה יותר ולאחריה הקצרה, כפי הדוגמא בהמשך.
16. הקשת שיפט ואנטר תתעלם מקיצורים המוגדרים בקובץ ini
17. פקודה שמוגדרת בקובץ ini שבתיקייה הנוכחית גוברת על ההגדרה שבקובץ ini שבתיקיית התג.
18. ניתן להשתמש בקובץ ini אחר מהברירת מחדל. עושים זאת ע"י שמגדירים שם קובץ ini אחר במשתנה הגלובלי zcl\_ini. ברירת מחדל ערך המשתנה הזה zcl\_ini.ini
- לכן אם תכתוב פקודה כזאת zcl\_ini := "z.ini" הסקריפט ישתמש בקובץ zcl\_ini.ini. ניתן ליצור סקריפט שייועד לכך שגם יריץ לאחור מכן את הסקריפט שורת הפקודה.
19. הסבר לקובץ ini המוצג כאן:

```

1 [alias]
2 @=func codeIndex(s) \r return asc(range(0,1,s)) + 256*asc(range(1,2,s)) \r end func \r
. setsel(end getsel()) \r findtag("code", codeindex("%1"), -1)
3 -=func codeIndex(s) \r return asc(range(0,1,s)) + 256*asc(range(1,2,s)) \r end func \r
. setsel(before getsel()) \r rfindtag("code", codeindex("%1"), -1)
4 #=setsel(end getsel()) \r findtag("ch", -1, asc("%1"))
5 /=setsel(end getsel()) \r find("%1", "%2")
6 :=run file "%1.prg"
7 ::=run file "%1.lib"
8 @@=setsel(end getsel()) \r findtag("code")
9
10 [options]
11 rohav=300
12 close_on_OK=0
13 words=-@,@@,::,/,#

```

- בשורה 1 זו מילת הקטע [alias]
- בשורה 2 מוגדר הקיצור @ שמחפש קוד. בהתחלה כתובה פונקציה ובהמשך הפקודה למקם את הסמן אחרי הבחירה אם ישנה, ואז לבצע חיפוש.
- בשורה 3 מוגדר הקיצור @- שמחפש קוד אחורה.
- בשורה 4 מוגדר הקיצור # שמחפש פרק קדימה. אם מוסיפים פרמטר אז הוא יחפש פרק עם יחוס של הפרמטר.
- בשורה 5 מוגדר הקיצור / שמחפש טקסט, אם מוסיפים פרמטר הוא יהיה תוי ג'וקר לשיטת החיפוש.
- בשורה 6 מוגדר הקיצור : שמריץ סקריפט בעל סיומת prg
- בשורה 7 מוגדר הקיצור :: שמריץ סקריפט בעל סיומת lib
- בשורה 8 מוגדר הקיצור @@ שמחפש קדימה כל קוד שהוא
- בשורה 10 זו מילת הקטע [options]
- בשורה 11 מוגדר רוחב תיבת השיחה של הסקריפט.
- בשורה 12 מוגדר שלא יסגור את חלון שורת הפקודה בהקשת אישור.
- בשורה 13 מוגדרות המילים שהסקריפט יבצע גם אם לא הוקש רווח אחרי הקיצור, וכמו שאמרנו הסדר כאן חשוב ולכן הוקדמו :: ל : ו@@ ל @

רעיונות יתקבלו בברכה