Dcl е съкращение от Dialog Control Language.

Dcl е описателен и интерпретативен език в AutoCAD. Обикновено се контролира от функциите на AutoLISP, за да взаимодейства с крайния потребител за събиране на информация и данни в средата на AutoCAD. Синтаксисът на Dcl се основава на дефиниране на основни плочки.

Плочките са елементите или групите от елементи в диалогов прозорец, като edit\_box или radio\_row. Диалоговият прозорец включва диалоговия прозорец и всички плочки в него. Подобно на други езици, които използват скоби "()" за затваряне на функции и сегменти от код, Dcl използва къдрави скоби "{}" като свой метод за затваряне на дефиниции на диалогови прозорци и свойствата и атрибутите на плочките.

Синтаксисът за атрибути и стойности е разделен със знак за равенство "=" и завършващ с точка и запетая ";". Например:

атрибут = стойност;

Dcl е изключително чувствителен към малки и големи букви език. **Всички имена в синтаксиса на езика трябва да са с малки букви.**

Употребата на всички етикети и ключове, дефинирани във файла Dcl от програмиста, трябва да съвпада с точната употреба на всички извиквания от AutoLISP за контрол и извличане на информация за плочки.

Например, ключ с име "Data\_Box" не може да бъде достъпен чрез "data\_BOX", тъй като случаят не е същият.

Също така имайте предвид, че текстовият низ за етикети за ключове може да съдържа само букви, цифри, тирета и долни черти. Както при AutoLISP, името на файла с разширение Dcl може да е различно от всяко от имената на дефинициите на диалозите в него. Ако дефиницията на диалоговия прозорец е копирана или преместена в друг Dcl файл, не забравяйте да промените извикването на load\_dialog към новото име на Dcl файл в рамките на функцията на AutoLISP. Също така, ако дефиницията на диалоговия прозорец се преименува, променете и името на диалоговия прозорец за извикването на new\_dialog. Dcl предоставя начин за проверка по време на процеса на проектиране и проверка за грешки. На audit\_level= 3; Dcl предоставя помощ и съвети и намира излишна информация за атрибути.

Пример 1.

Диалогов прозорец:

dcl\_settings : default\_dcl\_settings { audit\_level = 3; }

//---------------------------------------------------------------------------------------------------------

// MyFirst

//---------------------------------------------------------------------------------------------------------

MyFirst : dialog {

label = " Hello World";

spacer;

: text {

label = "This is my first dialog.";

alignment = centered;

}

spacer;

ok\_only;

}//MyFirst

Лисп код за диалоговия прозорец 1:

(defun c:MyFirst (/ Dcl\_Id%)

(princ "\nMyFirst")(princ)

; Load Dialog

(setq Dcl\_Id% (load\_dialog "C:/INCH6/DCL/MF.dcl"))

(new\_dialog "MyFirst" Dcl\_Id%)

; Dialog Actions

(start\_dialog)

; Unload Dialog

(unload\_dialog Dcl\_Id%)

(princ)

);defun c:MyFirst

Резултат:



Пример 2.

Диалогов прозорец 2

MyAlert1 : dialog {

key = "Title";

label = "";//Title$ from lsp file

spacer;

: text {

key = "Text1";

label = "";//Message$ from lsp file

width = 20.6;

alignment = centered;

}

spacer;

ok\_only;

}//MyAlert1

Код за диалоговия прозорец:

(defun MyAlert1 (Title$ Message$ / Dcl\_Id%)

(princ "\nMyAlert1")(princ)

; Load Dialog

(setq Dcl\_Id% (load\_dialog "MyDialogs.dcl"))

(new\_dialog "MyAlert1" Dcl\_Id%)

; Set Dialog Initial Settings

(set\_tile "Title" Title$)

(set\_tile "Text1" Message$)

; Dialog Actions

(start\_dialog)

; Unload Dialog

(unload\_dialog Dcl\_Id%)

(princ)

);defun MyAlert1

Обърнете внимание, че е дефинирана lisp функция, а не команда на autoCAD. Освен това функцията има два аргумента. Следователно трябва да се стартира със скоби и аргументи:

(MyAlert1 "My Message" "I’m going to figure this out.")

Резултат:



Синтаксисът на DCL файл е различен от синтаксиса на Visual LISP, защото дефинирате екран оформление, а не програма. Всеки обект в DCL файла има същия синтаксис:

[NAME ]: ITEM [ : ITEM …] { ATTRIBUTE = VALUE; … }

* NAME е нещо, което можете да използвате за препратка към плочката или диалоговия прозорец във вашето приложение. нормално, използвате NAME само когато дефинирате самия обект на диалогов прозорец.
* ITEM е името на определен тип плочки. Може да е резервно име или нещо по ваш собствен дизайн. Обикновено имате такъв ITEM, но можете да имате повече от един, ако е необходимо.
* ATTRIBUTE - Атрибутите за DCL обекта са между скобите.

Обикновено започваме с диалогова плочка и след това дефинират други плочки вътре в нея. В този случай са дефинирани три плочки (текст и два бутона). заедно с наименования обект ok\_cancel. Именуваният обект е a група от два бутона в ред, показващи текста OK и Cancel. Когато пишете DCL файлове, трябва да считате, че наименуваните обекти са единични плочки. Можете да създавате свои собствени наименувани обекти, като напр ok\_cancel във файла DCL, но основните, които ще искате, вече са предоставени в AutoCAD. ok\_cancel е често срещан, като осигурява два познати изхода от диалоговия прозорец.

****

HELLO1 : dialog {

label = "A Basic DCL";

: text {

key = "T1";

value="A Simple DCL example.";

}

: button {

key = "B1";

label = "Change 1";

}

: button {

key = "B2";

label = "Change 2";

}

ok\_cancel;

}

Всяка плочка в диалоговия прозорец има общ атрибут, наречен ключ, което е името, с което вашата програма ще препраща или разпознава конкретна плочка.

Стандартни видове плочки

Някои атрибути са общи за всички плочки, а други са уникални за конкретна плочка или имат специална значение само за някои плочки.

|  |  |
| --- | --- |
| **Tile** | **Description** |
| **boxed\_column** | Правоъгълно групиране на колони с граница. |
| **boxed\_radio\_column** | Правоъгълна област на колона за радио бутони с рамка. |
| **boxed\_radio\_row** | Правоъгълно групиране на редове за радио бутони с рамка. |
| **boxed\_row** | Правоъгълно групиране на редове с граница. |
| **button** | Бутон, като ОК. |
| **column** | Правоъгълно групиране на колони. |
| **edit\_box** | Поле за въвеждане на текст. |
| **errtile** | Областта на полето за отчитане на грешки в долната част на диалоговия прозорец. |
| **image** | **Graphic.** |
| **image\_button** | Графика, която се държи като бутон. |
| **list\_box** | Списък с избор на текст. |
| **paragraph** | Групиране на текстови\_части и обединяване. |
| **popup\_list** | Списък с избор на текст, който се изтегля надолу, за да се покаже списъкът; обикновено се показва само избрания елемент. |
| **radio\_button** | Бутон, свързан с други радио бутони в radio\_row или radio\_column, който може да бъде избран, за да посочи уникален избор. Само един радио бутон може да бъде избран наведнъж в радио ред или колона. Ако друг вече е избран, той автоматично ще бъде отменен. |
| **radio\_column** | Правоъгълно групиране на колони за радио бутони. |
| **radio\_row** | Правоъгълно групиране на редове за радио бутони. |
| **row** | Правоъгълно групиране на редове. |
| **slider** | Вертикална или хоризонтална плъзгаща се лента, използвана за обозначаване на стойност в рамките на диапазон. |
| **spacer** | Празно пространство между плочките. |
| **text** | Само текстово показване; без редактиране. |
| **text\_part** | Само текстово показване; част от параграф. |
| **toggle** | Бутон, който показва състояние на включване или изключване. Има отметка или квадратче с X в него. |

Атрибутите, които ще използвате най-често, са изброени в Таблица 11.2. Почти всички плочки имат такива достъпни за използване атрибути. Атрибутите етикет и стойност обаче не се използват с всички плочки. Например, стойността няма смисъл в случай на ред. Ако не искате да промените или модифицирате плочка, не се нуждае от ключ. Някои от атрибутите могат да бъдат променени по време на оценката на вашите Visual LISP програми използвайки подменюта, посочени в таблицата

Плочки с бутони

Плочката с бутони е област, върху която потребителят щраква. Можете да използвате бутони, за да позволите на оператора да излезе от диалогов прозорец за постоянно (като бутоните OK и Cancel), за да излезете от диалоговия прозорец за графики избор, за да промените стойностите в други плочки в текущия диалогов прозорец или да покажете друг диалогов прозорец кутия. Повечето програмисти спазват определени конвенции, когато използват бутони. Ако ще се използва бутон за графична селекция, символът по-малко от (<) често се поставя в края на етикета. Когато бутонът се използва за показване на друг диалогов прозорец, три точки (...) се показват в края на етикета.

Общи атрибути на плочки.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| **action** | String | Visual LISP зраз за оценка кога е избрана плочката или промяна. Това е едно от двете места, където можете да установите препратка към функцията за обратно извикване. За да презапишете препратката за обратно извикване на настройка, използвайте (ACTION\_TILE). |
| **alignment** | Left,  Right,  Centered | Подравняване на плочка в група. |
| **height** | Integer | Бройте в диалогови единици (знаци) на височината на плочката. |
| **key** | String | Име на плочката за външни препратки. |
| **is\_enabled** | True,  False | Може ли плочката да се избира и манипулира? По подразбиране, което е TRUE, може да бъде презаписан с помощта на (MODE\_TILE) subbr. |
| **is\_tab\_stop** | True,  False | Дали плочката е точка на спиране при табулиране през диалоговия прозорец кутия? |
| **label** | String | Низ, който да се показва с плочката. |
| **mnemomic** | String | Символ, който може да се използва за бързо достигане до плочката (горещо ключ). |
| **width** | Integer | Бройте в диалогови единици (знаци) на ширината на плочката. |
| **value** | String | Първоначална стойност на плочката. За да го презапишете, използвайте (SET\_TILE). |

**Пример 3.**

**Това е кода от лекцията**

**//DCL CODING STARTS HERE**

**test\_dcl2**

**: dialog**

**{**

**label = "Test Dialog No 2";**

**: edit\_box**

**{**

**label = "Enter Your Name :";**

**mnemonic = "N";**

**key = "name";**

**alignment = centered;**

**edit\_limit = 30;**

**edit\_width = 30;**

**}**

**: edit\_box**

**{**

**label = "Enter Your Age :";**

**mnemonic = "A";**

**key = "age";**

**alignment = centered;**

**edit\_limit = 3;**

**edit\_width = 3;**

**value = "";**

**}**

**: button**

**{**

**key = "accept";**

**label = "OK";**

**is\_default = true;**

**fixed\_width = true;**

**alignment = centered;**

**}**

**: errtile**

**{**

**width = 34;**

**}**

**}**

**//DCL CODING ENDS HERE**

И сега AutoLisp кодирането. Копирайте и поставете това и го запазете като "TEST\_DCL2.LSP" и след това заредете и стартирайте:

**;AUTOLISP CODING STARTS HERE**

**(prompt "\nType TEST\_DCL2 to run.....")**

**(defun C:TEST\_DCL2 ( / dcl\_id)**

**(setq dcl\_id (load\_dialog "C:/INCH6/DCL/Proba.dcl "))**

**(if (not (new\_dialog "test\_dcl2" dcl\_id))**

**(exit )**

**);if**

**(set\_tile "name" "Enter Name Here")**

**(mode\_tile "name" 2)**

**(action\_tile "name" "(setq name $value)")**

**(action\_tile "age" "(setq age $value)")**

**(action\_tile "accept" "(val1)")**

**(start\_dialog)**

**(unload\_dialog dcl\_id)**

**(alert (strcat "Your name is " name**

**"\nand you are " age " years of age."))**

**(princ)**

**);defun**

**-----------------------**

**(defun val1 ()**

**(if (= (get\_tile "name") "Enter Name Here")**

**(progn**

**(set\_tile "error" "You must enter a name!")**

**(mode\_tile "name" 2)**

**);progn**

**(val2)**

**);if**

**);defun**

**-------------------**

**(defun val2 ()**

**(if (< (atoi (get\_tile "age")) 1)**

**(progn**

**(set\_tile "error" "Invalid Age - Please Try Again!!")**

**(mode\_tile "age" 2)**

**);progn**

**(done\_dialog)**

**);if**

**);defun**

**(princ)**

**;AUTOLISP CODING ENDS HERE**