LEV\_COMP(6) 1996 LEV\_COMP(6)

NAME

lev\_comp - NetHack special levels compiler

SYNOPSIS

lev\_comp [ -w ] [ files ]

If no arguments are given, it reads standard input.

DESCRIPTION

Lev\_comp is a special level compiler for NetHack version 3.2

and higher. It takes description files as arguments and

produces level files that can be loaded by NetHack at run-

time.

The purpose of this tool is to provide NetHack administra-

tors and implementors with a convenient way for adding spe-

cial levels to the game, or modifying existing ones, without

having to recompile the entire world.

The -w option causes lev\_comp to perform extra checks on the

level and display extra warnings, however these warnings are

sometimes superfluous, so they are not normally displayed.

GRAMMAR

file : /\* nothing \*/

| levels

;

levels : level

| level levels

;

level : maze\_level

| room\_level

;

maze\_level : maze\_def flags lev\_init messages regions

;

room\_level : level\_def flags lev\_init messages rreg\_init rooms corridors\_def

;

level\_def : LEVEL\_ID ':' string

;

lev\_init : /\* nothing \*/

| LEV\_INIT\_ID ':' CHAR ',' CHAR ',' BOOLEAN ',' BOOLEAN ',' light\_state ',' walled

;

walled : BOOLEAN

| RANDOM\_TYPE

May Last change: 16 1

LEV\_COMP(6) 1996 LEV\_COMP(6)

;

flags : /\* nothing \*/

| FLAGS\_ID ':' flag\_list

;

flag\_list : FLAG\_TYPE ',' flag\_list

| FLAG\_TYPE

;

messages : /\* nothing \*/

| message messages

;

message : MESSAGE\_ID ':' STRING

;

rreg\_init : /\* nothing \*/

| rreg\_init init\_rreg

;

init\_rreg : RANDOM\_OBJECTS\_ID ':' object\_list

| RANDOM\_MONSTERS\_ID ':' monster\_list

;

rooms : /\* Nothing - dummy room for use with INIT\_MAP \*/

| roomlist

;

roomlist : aroom

| aroom roomlist

;

corridors\_def : random\_corridors

| corridors

;

random\_corridors: RAND\_CORRIDOR\_ID

;

corridors : /\* nothing \*/

| corridors corridor

;

corridor : CORRIDOR\_ID ':' corr\_spec ',' corr\_spec

| CORRIDOR\_ID ':' corr\_spec ',' INTEGER

;

corr\_spec : '(' INTEGER ',' DIRECTION ',' door\_pos ')'

;

aroom : room\_def room\_details

May Last change: 16 2

LEV\_COMP(6) 1996 LEV\_COMP(6)

| subroom\_def room\_details

;

subroom\_def : SUBROOM\_ID ':' room\_type ',' light\_state ',' subroom\_pos ',' room\_size ',' string roomfill

;

room\_def : ROOM\_ID ':' room\_type ',' light\_state ',' room\_pos ',' room\_align ',' room\_size roomfill

;

roomfill : /\* nothing \*/

| ',' BOOLEAN

;

room\_pos : '(' INTEGER ',' INTEGER ')'

| RANDOM\_TYPE

;

subroom\_pos : '(' INTEGER ',' INTEGER ')'

| RANDOM\_TYPE

;

room\_align : '(' h\_justif ',' v\_justif ')'

| RANDOM\_TYPE

;

room\_size : '(' INTEGER ',' INTEGER ')'

| RANDOM\_TYPE

;

room\_details : /\* nothing \*/

| room\_details room\_detail

;

room\_detail : room\_name

| room\_chance

| room\_door

| monster\_detail

| object\_detail

| trap\_detail

| altar\_detail

| fountain\_detail

| sink\_detail

| pool\_detail

| gold\_detail

| engraving\_detail

| stair\_detail

;

room\_name : NAME\_ID ':' string

;

room\_chance : CHANCE\_ID ':' INTEGER

May Last change: 16 3

LEV\_COMP(6) 1996 LEV\_COMP(6)

;

room\_door : DOOR\_ID ':' secret ',' door\_state ',' door\_wall ',' door\_pos

;

secret : BOOLEAN

| RANDOM\_TYPE

;

door\_wall : DIRECTION

| RANDOM\_TYPE

;

door\_pos : INTEGER

| RANDOM\_TYPE

;

maze\_def : MAZE\_ID ':' string ',' filling

;

filling : CHAR

| RANDOM\_TYPE

;

regions : aregion

| aregion regions

;

aregion : map\_definition reg\_init map\_details

;

map\_definition : NOMAP\_ID

| map\_geometry MAP\_ID

;

map\_geometry : GEOMETRY\_ID ':' h\_justif ',' v\_justif

;

h\_justif : LEFT\_OR\_RIGHT

| CENTER

;

v\_justif : TOP\_OR\_BOT

| CENTER

;

reg\_init : /\* nothing \*/

| reg\_init init\_reg

;

init\_reg : RANDOM\_OBJECTS\_ID ':' object\_list

| RANDOM\_PLACES\_ID ':' place\_list

May Last change: 16 4

LEV\_COMP(6) 1996 LEV\_COMP(6)

| RANDOM\_MONSTERS\_ID ':' monster\_list

;

object\_list : object

| object ',' object\_list

;

monster\_list : monster

| monster ',' monster\_list

;

place\_list : place

| place ',' place\_list

;

map\_details : /\* nothing \*/

| map\_details map\_detail

;

map\_detail : monster\_detail

| object\_detail

| door\_detail

| trap\_detail

| drawbridge\_detail

| region\_detail

| stair\_region

| portal\_region

| teleprt\_region

| branch\_region

| altar\_detail

| fountain\_detail

| mazewalk\_detail

| wallify\_detail

| ladder\_detail

| stair\_detail

| gold\_detail

| engraving\_detail

| diggable\_detail

| passwall\_detail

;

monster\_detail : MONSTER\_ID chance ':' monster\_c ',' m\_name ',' coordinate

monster\_infos

;

monster\_infos : /\* nothing \*/

| monster\_infos monster\_info

;

monster\_info : ',' string

| ',' MON\_ATTITUDE

| ',' MON\_ALERTNESS

May Last change: 16 5

LEV\_COMP(6) 1996 LEV\_COMP(6)

| ',' alignment

| ',' MON\_APPEARANCE string

;

object\_detail : OBJECT\_ID object\_desc

| COBJECT\_ID object\_desc

;

object\_desc : chance ':' object\_c ',' o\_name ',' object\_where object\_infos

;

object\_where : coordinate

| CONTAINED

;

object\_infos : /\* nothing \*/

| ',' curse\_state ',' monster\_id ',' enchantment optional\_name

| ',' curse\_state ',' enchantment optional\_name

| ',' monster\_id ',' enchantment optional\_name

;

curse\_state : RANDOM\_TYPE

| CURSE\_TYPE

;

monster\_id : STRING

;

enchantment : RANDOM\_TYPE

| INTEGER

;

optional\_name : /\* nothing \*/

| ',' NONE

| ',' STRING

;

door\_detail : DOOR\_ID ':' door\_state ',' coordinate

;

trap\_detail : TRAP\_ID chance ':' trap\_name ',' coordinate

;

drawbridge\_detail: DRAWBRIDGE\_ID ':' coordinate ',' DIRECTION ',' door\_state

;

mazewalk\_detail : MAZEWALK\_ID ':' coordinate ',' DIRECTION

;

wallify\_detail : WALLIFY\_ID

;

May Last change: 16 6

LEV\_COMP(6) 1996 LEV\_COMP(6)

ladder\_detail : LADDER\_ID ':' coordinate ',' UP\_OR\_DOWN

;

stair\_detail : STAIR\_ID ':' coordinate ',' UP\_OR\_DOWN

;

stair\_region : STAIR\_ID ':' lev\_region ',' lev\_region ',' UP\_OR\_DOWN

;

portal\_region : PORTAL\_ID ':' lev\_region ',' lev\_region ',' string

;

teleprt\_region : TELEPRT\_ID ':' lev\_region ',' lev\_region teleprt\_detail

;

branch\_region : BRANCH\_ID ':' lev\_region ',' lev\_region

;

teleprt\_detail : /\* empty \*/

| ',' UP\_OR\_DOWN

;

lev\_region : region

| LEV '(' INTEGER ',' INTEGER ',' INTEGER ',' INTEGER ')'

;

fountain\_detail : FOUNTAIN\_ID ':' coordinate

;

sink\_detail : SINK\_ID ':' coordinate

;

pool\_detail : POOL\_ID ':' coordinate

;

diggable\_detail : NON\_DIGGABLE\_ID ':' region

;

passwall\_detail : NON\_PASSWALL\_ID ':' region

;

region\_detail : REGION\_ID ':' region ',' light\_state ',' room\_type prefilled

;

altar\_detail : ALTAR\_ID ':' coordinate ',' alignment ',' altar\_type

;

gold\_detail : GOLD\_ID ':' amount ',' coordinate

;

engraving\_detail: ENGRAVING\_ID ':' coordinate ',' engraving\_type ',' string

;

May Last change: 16 7

LEV\_COMP(6) 1996 LEV\_COMP(6)

monster\_c : monster

| RANDOM\_TYPE

| m\_register

;

object\_c : object

| RANDOM\_TYPE

| o\_register

;

m\_name : string

| RANDOM\_TYPE

;

o\_name : string

| RANDOM\_TYPE

;

trap\_name : string

| RANDOM\_TYPE

;

room\_type : string

| RANDOM\_TYPE

;

prefilled : /\* empty \*/

| ',' FILLING

| ',' FILLING ',' BOOLEAN

;

coordinate : coord

| p\_register

| RANDOM\_TYPE

;

door\_state : DOOR\_STATE

| RANDOM\_TYPE

;

light\_state : LIGHT\_STATE

| RANDOM\_TYPE

;

alignment : ALIGNMENT

| a\_register

| RANDOM\_TYPE

;

altar\_type : ALTAR\_TYPE

| RANDOM\_TYPE

;

May Last change: 16 8

LEV\_COMP(6) 1996 LEV\_COMP(6)

p\_register : P\_REGISTER '[' INTEGER ']'

;

o\_register : O\_REGISTER '[' INTEGER ']'

;

m\_register : M\_REGISTER '[' INTEGER ']'

;

a\_register : A\_REGISTER '[' INTEGER ']'

;

place : coord

;

monster : CHAR

;

object : CHAR

;

string : STRING

;

amount : INTEGER

| RANDOM\_TYPE

;

chance : /\* empty \*/

| PERCENT

;

engraving\_type : ENGRAVING\_TYPE

| RANDOM\_TYPE

;

coord : '(' INTEGER ',' INTEGER ')'

;

region : '(' INTEGER ',' INTEGER ',' INTEGER ',' INTEGER ')'

;

NOTE:

Lines beginning with '#' are considered comments.

The contents of a "MAP" description of a maze is a rectangle

showing the exact level map that should be used for the

given part of a maze. Each character in the map corresponds

to a location on the screen. Different location types are

denoted using different ASCII characters. The following

characters are recognized. To give an idea of how these are

used, see the EXAMPLE, below. The maximum size of a map is

May Last change: 16 9

LEV\_COMP(6) 1996 LEV\_COMP(6)

normally 76 columns by 21 rows.

'-' horizontal wall

'|' vertical wall

'+' a doorway (state is specified in a DOOR declaration)

'A' open air

'B' boundary room location (for bounding unwalled irregular regions)

'C' cloudy air

'I' ice

'S' a secret door

'H' a secret corridor

'{' a fountain

'\' a throne

'K' a sink (if SINKS is defined, else a room location)

'}' a part of a moat or other deep water

'P' a pool

'L' lava

'W' water (yes, different from a pool)

'T' a tree

'F' iron bars

'#' a corridor

'.' a normal room location (unlit unless lit in a REGION declaration)

' ' stone

EXAMPLE

Here is an example of a description file (a very simple

one):

MAZE : "fortress", random

GEOMETRY : center , center

MAP

}}}}}}}}}

}}}|-|}}}

}}|-.-|}}

}|-...-|}

}|.....|}

}|-...-|}

}}|-.-|}}

}}}|-|}}}

}}}}}}}}}

ENDMAP

MONSTER: '@', "Wizard of Yendor", (4,4)

OBJECT: '"', "Amulet of Yendor", (4,4)

# a hell hound flanking the Wiz on a random side

RANDOM\_PLACES: (4,3), (4,5), (3,4), (5,4)

MONSTER: 'd', "hell hound", place[0]

# a chest on another random side

OBJECT: '(', "chest", place[1]

# a sack on a random side, with a diamond and maybe a ruby in it

CONTAINER: '(', "sack", place[2]

OBJECT: '\*', "diamond", contained

OBJECT[50%]: '\*', "ruby", contained

May Last change: 16 10

LEV\_COMP(6) 1996 LEV\_COMP(6)

# a random dragon somewhere

MONSTER: 'D', random, random

# 3 out of 4 chance for a random trap in the EAST end

TRAP[75%]: random, (6,4)

# an electric eel below the SOUTH end

MONSTER: ';', "electric eel", (4,8)

# make the walls non-diggable

NON\_DIGGABLE: (0,0,8,8)

TELEPORT\_REGION: levregion(0,0,79,20), (0,0,8,8)

This example will produce a file named "fortress" that can

be integrated into one of the numerous mazes of the game.

Note especially the final, TELEPORT\_REGION specification.

This says that level teleports or other non-stairway

arrivals on this level can land anywhere on the level except

the area of the map. This shows the use of the ``levre-

gion'' prefix allowed in certain region specifications.

Normally, regions apply only to the most recent MAP specifi-

cation, but when prefixed with ``levregion'', one can refer

to any area of the level, regardless of the placement of the

current MAP in the level.

AUTHOR

Jean-Christophe Collet, David Cohrs.

SEE ALSO

dgn\_comp(6), nethack(6)

BUGS

Probably infinite. Most importantly, still needs additional

bounds checking.

May Last change: 16 11