# Rapid LP Development with Mosel

Tom Halliburton
Energy Modeling Consultants Ltd.

for

Operations Research Society of New Zealand
November, 2003

# Building Linear Programs

- Interface options
- Dash Optimisation
- Dash product range
- Main features of Mosel
- Example problem

# High Level Math Programming Interfaces

- Modeling Languages
  - GAMS
  - AMPL
  - LINGO
  - MPL
- Excel Solver
- Programming APIs
  - ILOG Planner & Concert
  - Xpress BCL
- Recent Modeling Languages
  - ILOG OPL
  - Xpress Mosel

# Dash Optimisation

- Owned by Bob Daniels & Robert Ashford
- Offices in UK and US
- Established in 1980s
- Solvers available:
  - LP primal, dual, interior point
  - MIP
  - Quadratic Programming and MIP + QP
  - Successive approximations LP, with MIP
- Programming API for C, VB, Java - BCL
- Modeling language - Mosel

# Mosel Overview

- Developed by Dash, in France
- IVE development environment written in New Jersey
- First released 2001
- Interfaces directly with all Dash solvers
  - not via files
- More like a programming language than AMPL
- Graphical development environment for Windows
  - edit, compile, run, debug within the IDE
- Command line interface for other operating systems
- Provides a compiled model file for distribution
  - preserves intellectual property
  - facilitates run time licensing
- Can be embedded in C, VB, Java code

# Why Mosel?

- Xpress solvers amongst the best
- Price competitive
  - Mosel, IVE, LP, MIP:  US$9900
- Flexibility of licensing
- Good telephone & email support
- Successful development using Mosel v1.0.1
  - early versions of software sometimes risky
  - no gotchas
- Licensing procedures usually straight forward
- Documentation good
- Stick with something I trust
  - time required to learn new systems, languages, etc

# Programming Constructs

- forall loop
- while
- if then else
- case
- repeat until
- functions & subroutines
- sets & set operations
- read from text, spreadsheets or databases
- dynamic and sparse arrays, with “exists” function

# The Burglar Problem

```
declarations
 Items={"camera", "necklace", "vase", "picture", "tv", "video",
  "chest", "brick"}                    ! Index set for items
  VALUE: array(Items) of real          ! Value of items
  WEIGHT: array(Items) of real         ! Weight of items
  WTMAX=102                            ! Max weight allowed
  x: array(Items) of mpvar             ! 1 if we take item i; 0 otherwise
end-declarations

VALUE := [15, 100, 90, 60, 40, 15, 10,  1]
 WEIGHT:= [ 2,  20, 20, 30, 40, 30, 60, 10]

MaxVal:= sum(i in Items) VALUE(i)*x(i)
WtMax:= sum(i in Items) WEIGHT(i)*x(i) <= WTMAX
forall(i in Items) x(i) is_binary

maximize(MaxVal)
```

# IVE - Development Environment

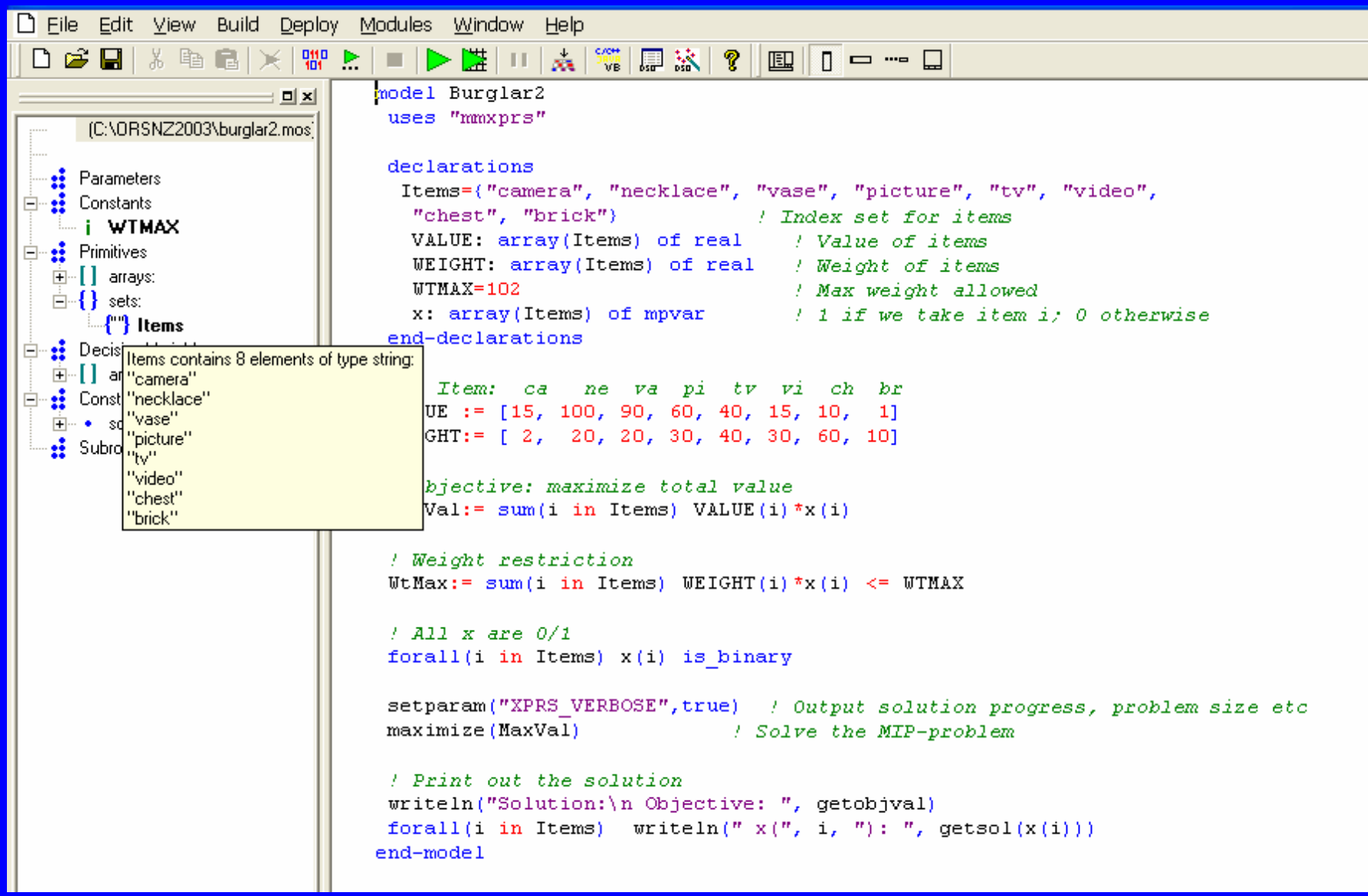
File Edit View Build Deploy Modules Window Help

(C:\ORSNZ2003\burglar2.mos)

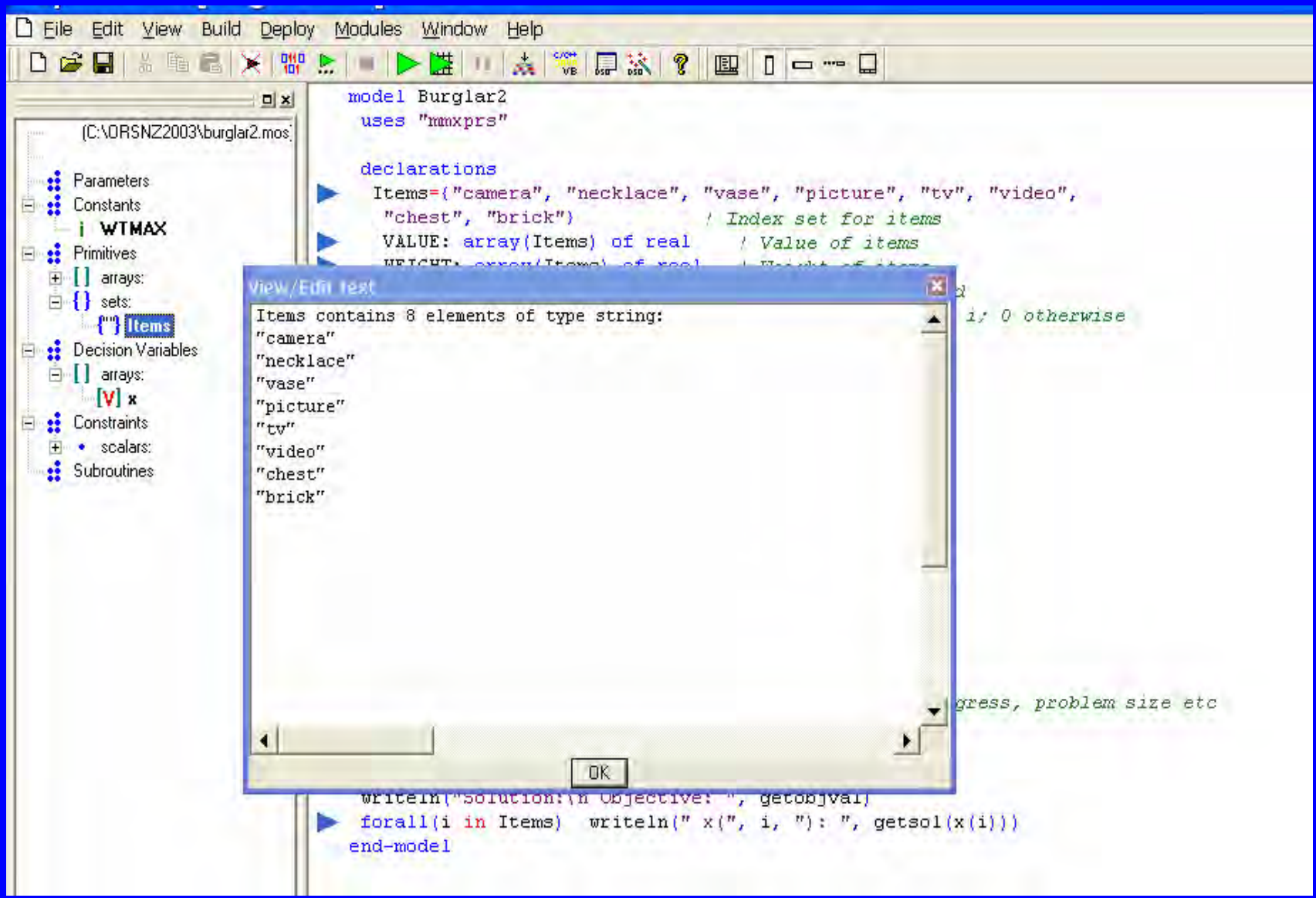
- Parameters
- Constants
  - WTMAX
- Primitives
  - [] arrays:
  - {} sets:
    - {"} Items
- Decision Variables
  - [] arrays:
- Constraints
  - • scalars:
- Subroutines

```
model Burglar2
 uses "mmxprs"

 declarations
  Items={"camera", "necklace", "vase", "picture", "tv", "video",
   "chest", "brick"}            ! Index set for items
   VALUE: array(Items) of real    ! Value of items
   WEIGHT: array(Items) of real   ! Weight of items
   WTMAX=102                      ! Max weight allowed
   x: array(Items) of mpvar       ! 1 if we take item i; 0 otherwise
 end-declarations

 !   Item:  ca   ne  va  pi  tv  vi  ch  br
 VALUE := [15, 100, 90, 60, 40, 15, 10,  1]
 WEIGHT:= [ 2,  20, 20, 30, 40, 30, 60, 10]

 ! Objective: maximize total value
 MaxVal:= sum(i in Items) VALUE(i)*x(i)

 ! Weight restriction
 WtMax:= sum(i in Items) WEIGHT(i)*x(i) <= WTMAX

 ! All x are 0/1
 forall(i in Items) x(i) is_binary

 setparam("XPRS_VERBOSE",true)  ! Output solution progress, problem size etc
 maximize(MaxVal)             ! Solve the MIP-problem

 ! Print out the solution
 writeln("Solution:\n Objective: ", getobjval)
 forall(i in Items)  writeln(" x(", i, "): ", getsol(x(i)))
end-model
```

# Viewing Data - move mouse over name



# Viewing Data

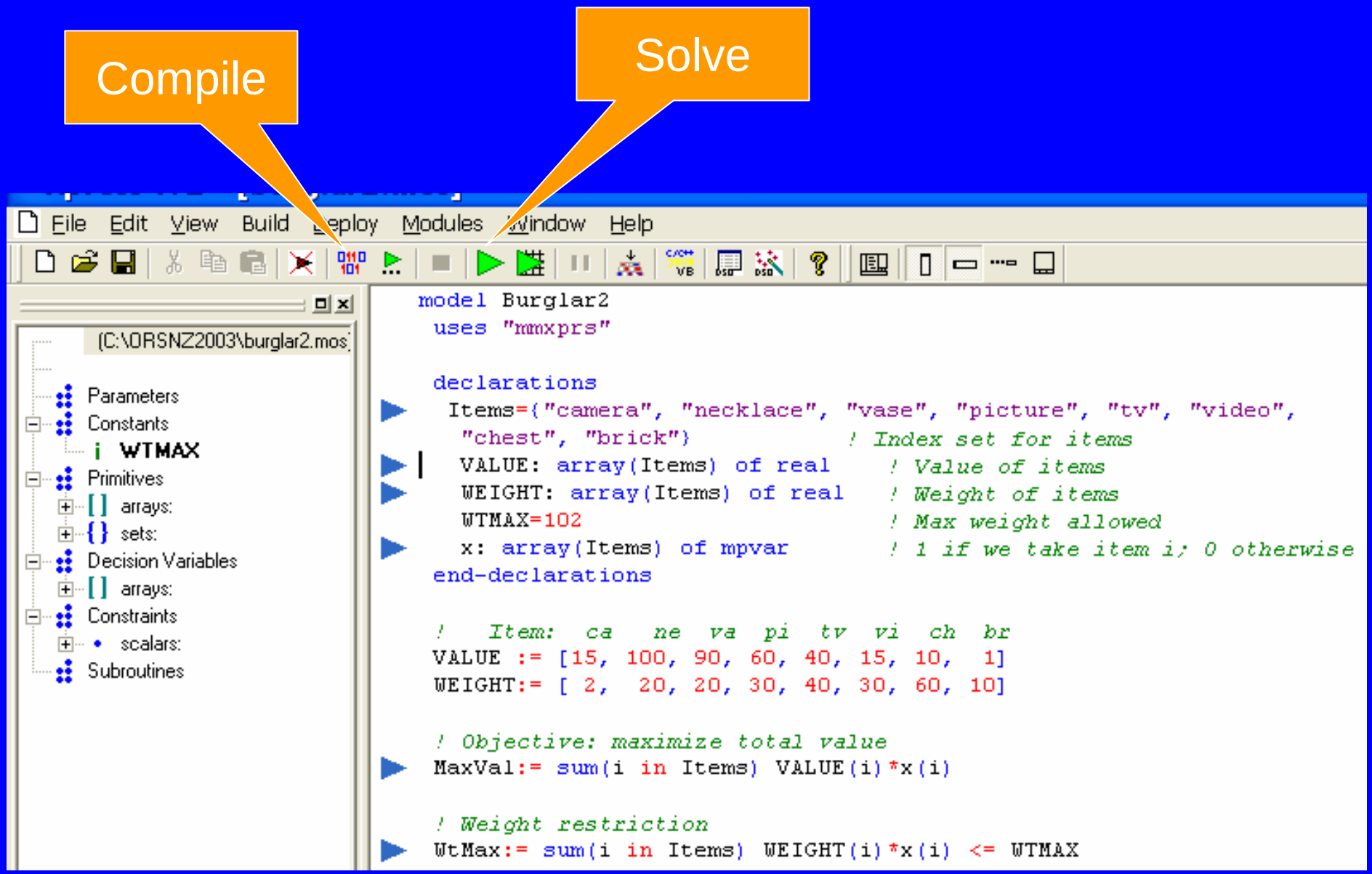


# Syntax Error

File Edit View Build Deploy Modules Window Help

```
!   Item:  ca   ne  va  pi  tv  vi  ch  br
VALUE := [15, 100, 90, 60, 40, 15, 10,  1]
WEIGHT:= [ 2,  20, 20, 30, 40, 30, 60, 10]

! Objective: maximize total value
MaxVal:= sum(i in Items) VALUE(i)*x(i)

! Weight restriction
WtMax:= sum(i in Items) WEIGHT(i)*x(i) <= WTAX

! All x are 0/1
forall(i in Items) x(i) is_binary

setparam("XPRS_VERBOSE",true)   ! Output solution progress, problem size etc
maximize(MaxVal)                ! Solve the MIP-problem

! Print out the solution
writeln("Solution:\n Objective: ", getobjval)
forall(i in Items)  writeln(" x(", i, "): ", getsol(x(i)))
end-model
```

```
C:\ORSNZ2003\burglar2.mos(Line:  21, Col:  47) : error 123: 'WTAX' is not defined
C:\ORSNZ2003\burglar2.mos(Line:  21, Col:  47) : error 123: 'WtMax' is not defined
C:\ORSNZ2003\burglar2.mos(Line:  21, Col:  47) : error 123: 'WTAX' is not defined
```


Compile
Solve
File Edit View Build Deploy Modules Window Help
(C:\ORSNZ2003\burglar2.mos)
Parameters
Constants
WTMAX
Primitives
arrays:
sets:
Decision Variables
arrays:
Constraints
scalars:
Subroutines
model Burglar2
uses "mmxprs"
declarations
Items={"camera", "necklace", "vase", "picture", "tv", "video",
"chest", "brick"} ! Index set for items
VALUE: array(Items) of real ! Value of items
WEIGHT: array(Items) of real ! Weight of items
WTMAX=102 ! Max weight allowed
x: array(Items) of mpvar ! 1 if we take item i; 0 otherwise
end-declarations
! Item: ca ne va pi tv vi ch br
VALUE := [15, 100, 90, 60, 40, 15, 10, 1]
WEIGHT:= [ 2, 20, 20, 30, 40, 30, 60, 10]
! Objective: maximize total value
MaxVal:= sum(i in Items) VALUE(i)*x(i)
! Weight restriction
WtMax:= sum(i in Items) WEIGHT(i)*x(i) <= WTMAX

Text output from Mosel/Optimizer. View/Edit

```
Reading Problem xprs_e24_l3eae68
Problem Statistics
          1 (    108 spare) rows
          8 (      0 spare) structural columns
          8 (   3432 spare) non-zero elements
Global Statistics
          8 entities        0 sets        0 set members
Presolved problem has:      1 rows       7 cols       7 non-zeros

   Its         Obj Value      S   Ninf  Nneg        Sum Inf  Time
     0         15.000000      D      0     0        .000000     0
     1        295.000000      D      0     0        .000000     0
Optimal solution found

Generating cuts

   Its         Obj Value    Type    Cuts Added / Deleted    Time
     1        280.000000      K            1         0         0

Cuts in the matrix         : 1
Cut elements in the matrix : 5

Cuts in the cutpool        : 1
Cut elements in the cutpool: 5

    Node       Sols   Best Solution     Best Bound  Active    Time
 *     1          1      280.000000     280.000000       0       0 *
 *** Search completed ***     Time:     0 Nodes:     1
Number of integer feasible solutions found is 1
Best integer solution found is   280.000000
Solution:
 Objective: 280
 x(camera): 1
 x(necklace): 1
 x(vase): 1
 x(picture): 1
 x(tv): 0
 x(video): 1
 x(chest): 0
 x(brick): 0
```

# Solving Within a Loop

```
maximize(MaxVal)
forall(loot in Items) do
      Fixit:= x(loot)=1
      maximize(MaxVal)
      writeln("\nMust steal ", loot, "  Objective: ",getobjval)
      forall(i in Items | getsol(x(i))=1 ) write("  ",Items(i))
end-do

maximize(XPRS_LIN,MaxVal)
forall (loot in Items) write(loot,": ",getsol(x(loot)),"  ")
writeln("\nValue of a stronger sack = ",getdual(WtMax))
```

# Output from Example

Objective: 280
Must steal camera  Objective: 280  camera  necklace  vase  picture  video
……..
Must steal tv  Objective: 246  camera  necklace  vase  tv  brick
……..
Must steal chest  Objective: 215  camera  necklace  vase  chest

Optimal values of variables from MIP:
camera: 1  necklace: 1  vase: 1  picture: 1  tv: 0  video: 0  chest: 0  brick: 1

Solve as a relaxed LP only  Objective: 286
camera: 1  necklace: 1  vase: 1  picture: 1  tv: 0.5  video: 0  chest: 0  brick: 1
Value of a stronger sack = 1


optimized by
Xpress MP