MasterPlan 1.7.5 Documentation 2005-02-23

Anders Forslund

Index

1 Purpose

Masterplan is a Win32(9x/NT), utility designed primary for gamemasters of roleplaying games.
The MasterPlan software can for be used to create events,people, places, monsters, treasures, items etc etc with unique attributes

You can create custom scripts utilizing Your tables and generate for instance whole characters to any game system. Entire adventures can be generated this way...

Simulate complete combat encounters with custom scripts

What You need to do in order to use the MasterPlan software is tables with data. The software is shipped with a number of example files, but the main work has to be done by Yourself, this way, the only limit of what you can do with the software is Your own imagination.

2 MasterPlan Features

Table lookup, both fixed and random
Scripts
Reference tables and scripts
Multiple and relative referencing
Recursive referencing
Dice functions(replaces dicesets with results)
Printing and exporting functions
Script conditional control flow.
Create invisible variables to save generated values. These values can be used later
in the script or by tables. Variables can be numbers or characters.
Invisible variables can be used in table headers, affecting the outcome
from a specific table. No duplicate of entries in the the same instance unless there is a limited amount of entries - especially useful in scripts.
Treeview explorer for easy browsing of resource files
Mathematical expressions
  +,-,*,/,\,^,%(MOD)
Mathematical functions
  Cos,Sin,Tan,Sqr,Exp,Log
  Rnd(Round),Val(converts strings to numbers)
  PI constant,Now constant(returns current system date and time))
Other functions
  Get(string) gets user input, string is the text that will be displayed.
The mathematical expressions/functions can be used in conjunction with
invisible variables.

Table Editor
Script Editor
Code completion for references
Dice Probablility Analysis(Shadowrun, Hjältesystem)

3 System Requirements

The software is tested on NT4,Win2k,W98SE and should have no problem running on any machine capable of running these operating systems.

4 About the goals of MasterPlan

The first inspiration for MasterPlan came from a truly amazing RPG sourcebooks, the Central Casting-series from Task Force Games(http://www.task-force-games.com) , of course then we could see all sorts of possibillites!
The MasterPlan software have been developed with the basic purpose of that it should be possible to fully recreate the Central Castings-books and we truly believes it is possible to do this.

5 About AnderWare

AndersWare consists of roleplayers who strive to ease the burden for gamemasters. The MasterPlan software exists because we wanted a useful software to help our gamemasters, that means that resourcefiles will continually updated as we create new ones. If You make a good resourcefile that You think other gamers may find useful, please submit it to us and we will publish it on our website.

The software is Freeware, running it is at Your own risk. AnderWare takes no responsibility for any damages to your computers hard- or software the MasterPlan software may cause.

6 About this manual

Some screenshots are taken from earlier versions of the software and thus maybe not entirely correct. However, all screenshots used in this document depicts the topic that is covered.

7 Installation and Configuration

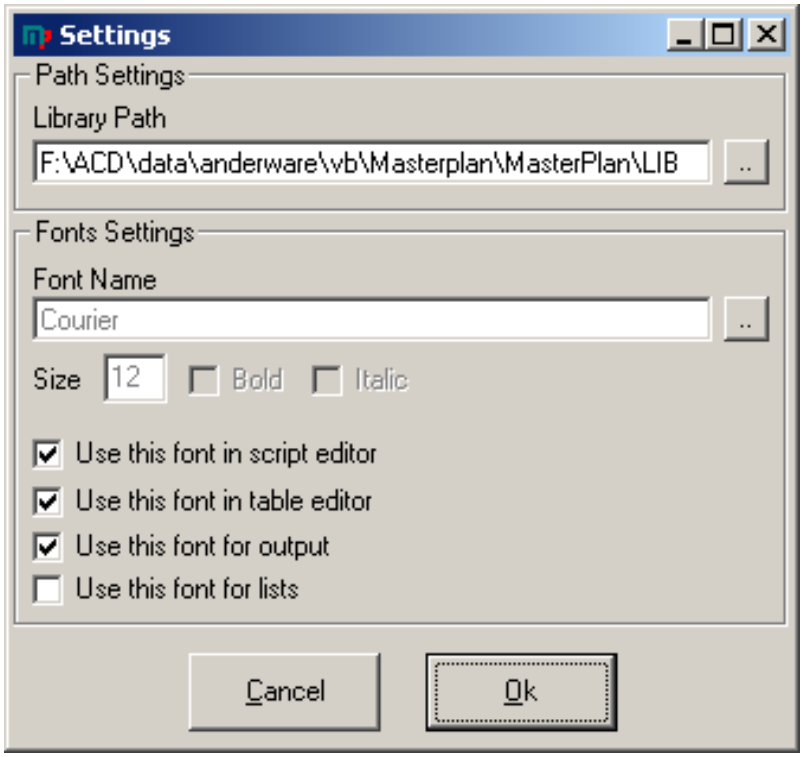
7.1 Installation

Simply run the supplied setup.exe from your file manager of choice (I recommend Total Commander but Windows Explorer works as well.)

7.2 Configuration

MasterPlan is set to work with a minimum of configuration. They are found in tne menu File\Settings

This will bring upp the Settings dialog.



7.3 Library Path

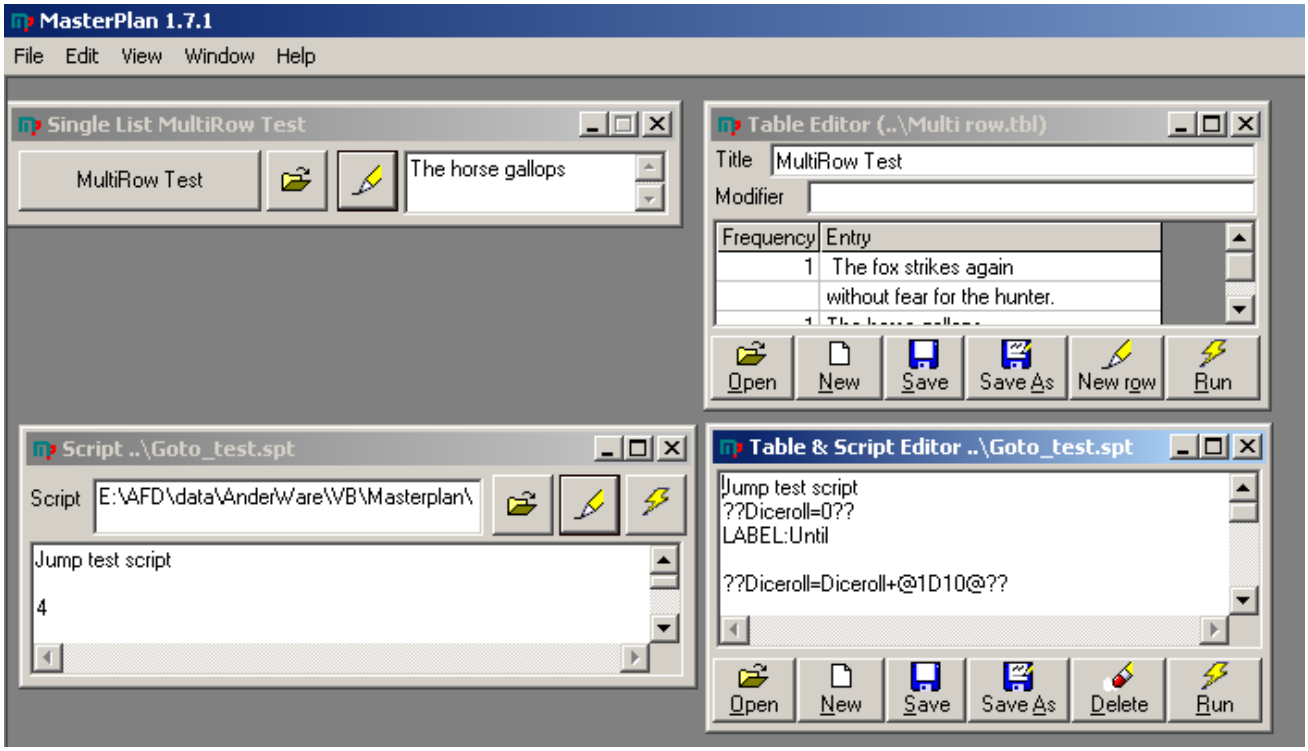
In order to make it possible for the software(the MasterPlan Explorer viewern in particular) to work properly it is neccessary to set the path to the resource files folder.

7.4 Font Settings

To enhance the readability of scripts etc is it possible to select another font that suits your taste.
It is recommended to use a fixed width font such as Courier for this.

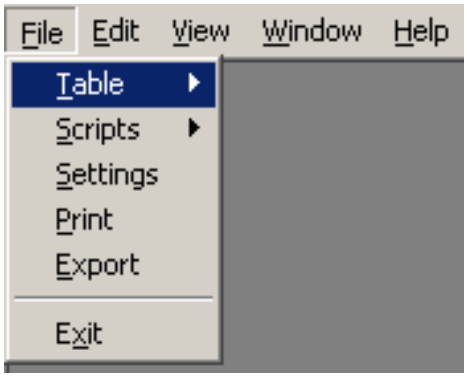
8 Graphical Interface

MasterPlan uses standard Windows MDI(Multiple Document Interface), meaning of course that you can have several smaller windows open at the same time.



8.1 Keyboard Shortcuts

All menus kan be reached by keyboard commands. On these screenshots the actual commands are not shown. Simply hold the ALT-key down and the shorrtcuts will appear.

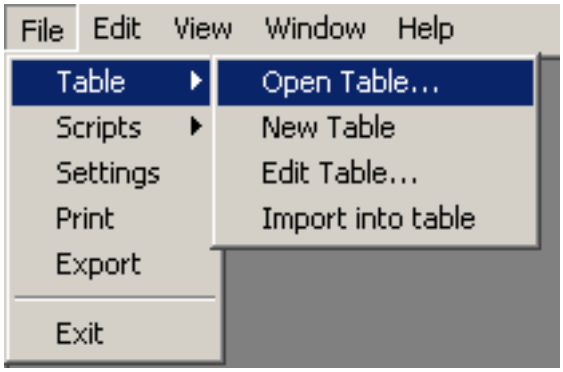


8.2 Tooltips

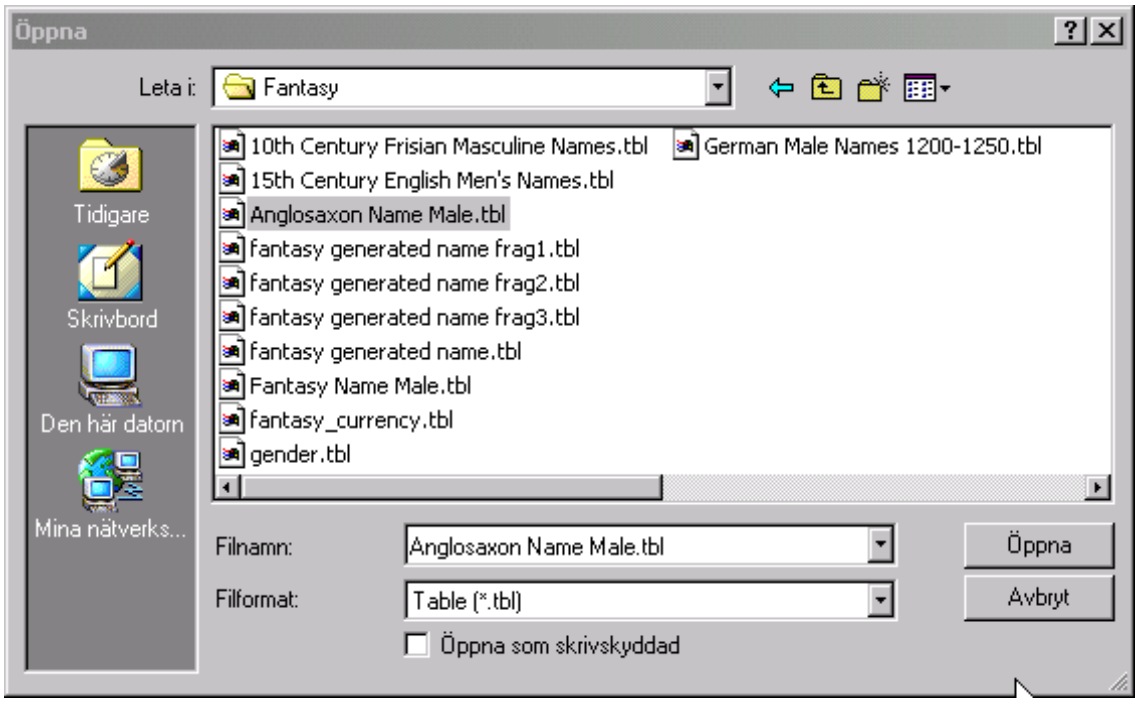
For graphical buttons, a tooltip will appear with a description of the button when the cursor is positioned over the button for a few seconds.

8.3 Open File Dialog

Three dots(...) always signifies the File Open Dialog and can be found in menues



or on buttons also used is the familiar folder icon.

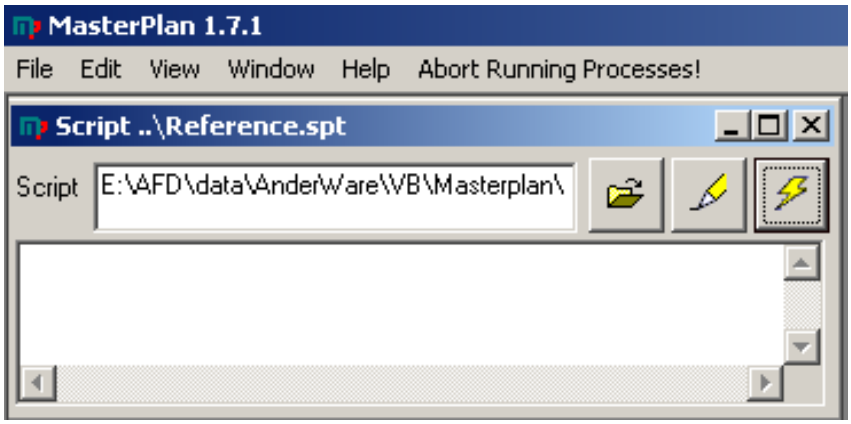


8.4 Printing /exporting data

All generated output can be printed to a Windows printer and also exported into ASCII text file. It is also possible to print/export tables and scripts.

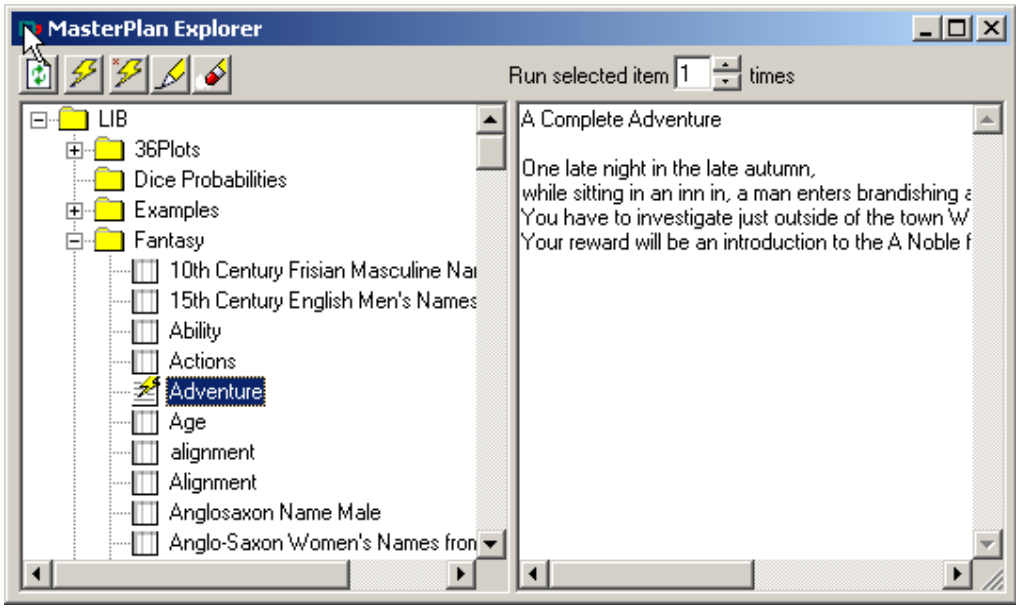
8.5 Aborting running processes

It is quite possible to accidently design tables and scripts that can get stuck in an infinite loop (a table calling itself over and over for example). When a table or script is being processed a special menu item appears, the “Abort Running Processes!”. Clicking this will stop processes running.

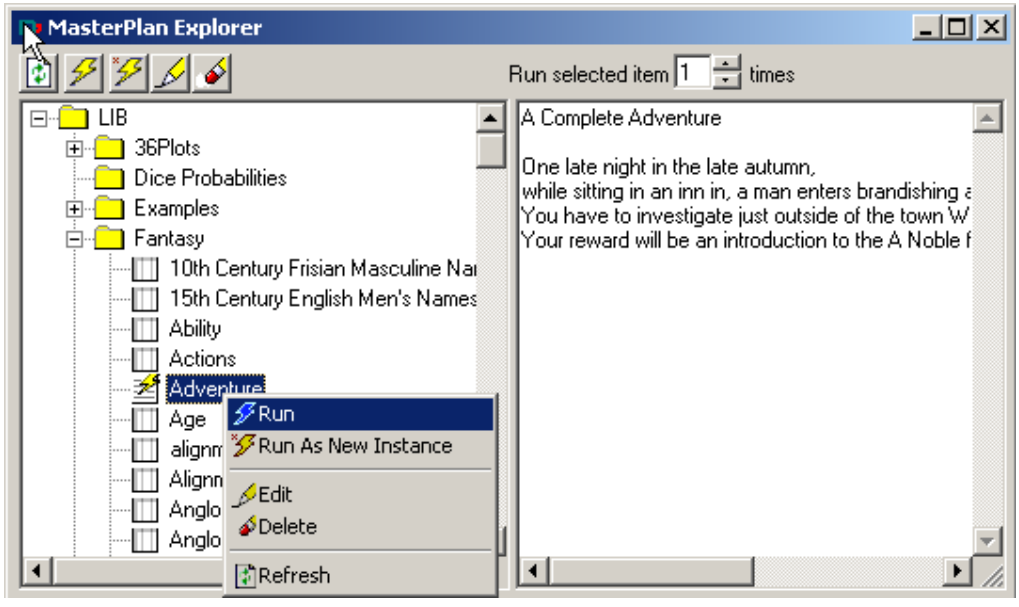


9 MasterPlan Explorer

In order to make it easier to browse and run the resource files, a treeview explorer is supplied.
The files showed here are the legit resource files found(recursively) in the Library Path (see above on how to change the Library Path).
By simply clicking the resource file(table or script) the result is shown in the output window to the right.



It is also possible to rightclick a resource and select an alternative from the popup menu that appears.



9.1.1 Run

Returns the resource file result in the output window to the right(same as leftclicking the resource file)

9.1.2 Run As New Instance

Opens up a Single List window if it is a table or a Script window(if it is a script) with the resource file.

9.1.3 Edit

Opens up a Edit Table window if it is a table or an Edit Script window(if it is a script) with the resource file.

9.1.4 Delete

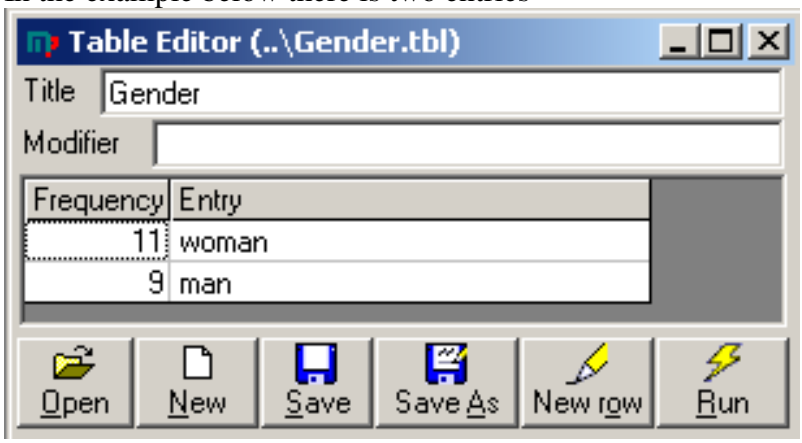
Deletes the resource after checking with the user.

9.1.5 Refresh

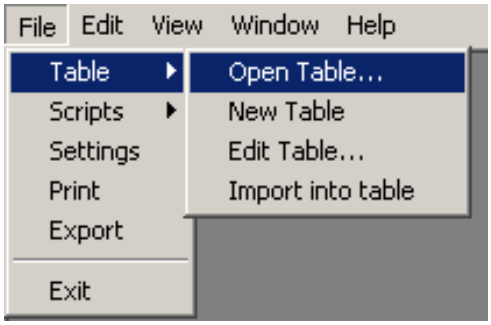
Rereads the library path and load the valid resource files into the Explorer

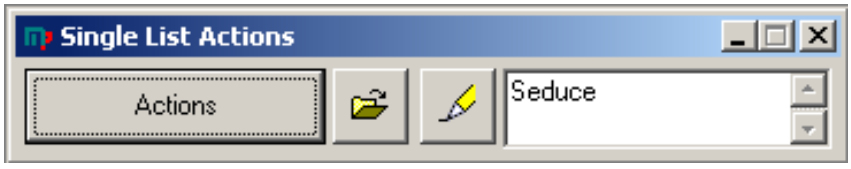
10 Tables

The heart of MasterPlan!
Each table contains a number of entries. Each entry consists of a text and a number describing that entrys probability(frequency) of the total in that table.
In the example below there is two entries



10.1 Running Tables

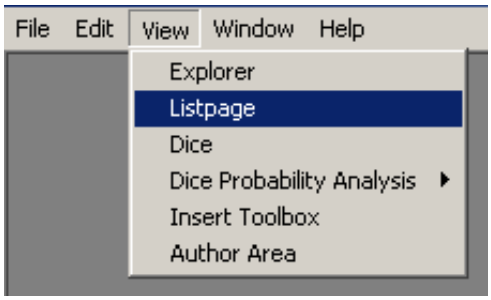


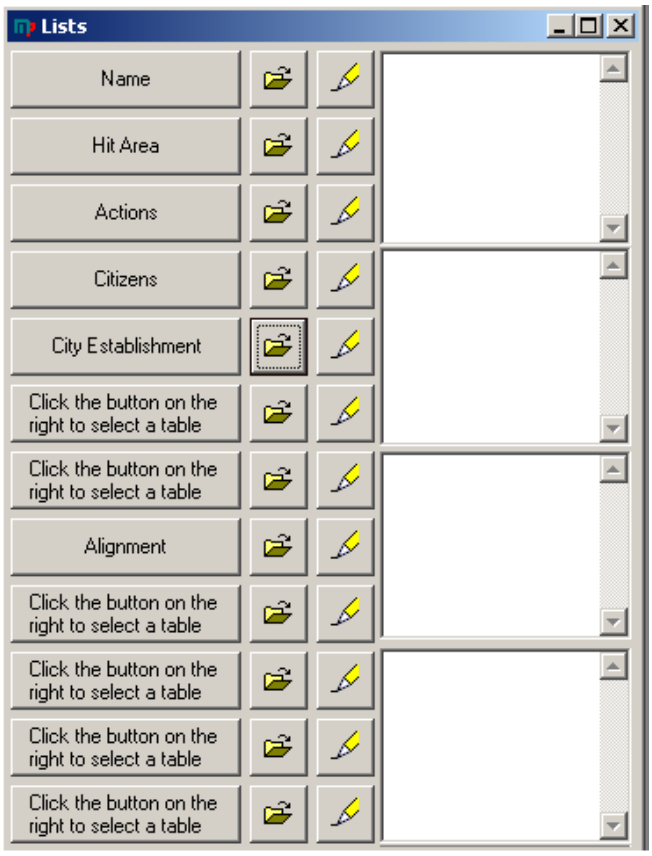


Each click on the name button generates new data. The Open file-button opens up the file open dialog. It is also possible to open up the table in editor-mode by clicking the Edit file -button

10.2 Listpage

Sometimes You probably want to be able to run several tables at once and thats what the Listpage are for!





10.3 Editing Tables

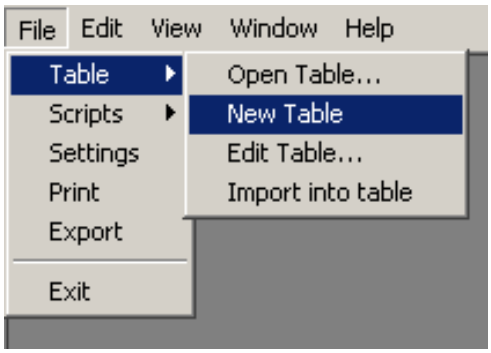
Each text can reference other tables, including the same table

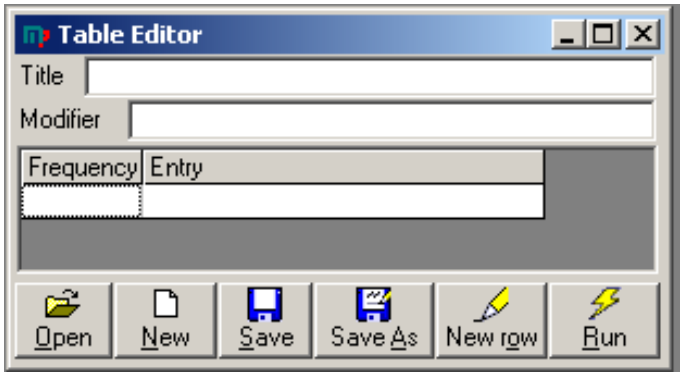
The naming of tables are of course important in order to avoid duplicates and hardtraced errors.
The tablename is not casesensitive, but when you make a reference
to it, it is important to make it in the right case for the the
the right desired result.

The more detail you put into the table files the more accurate results will be generated. For instance, a list of the names and titles of the nobles in your country instead of randomly generated nobles.

One approach would be to use the MasterPlan software when You create Your world/campaign from the beginning.
That way You will have accurate information about Your campaign right at Your fingertips thereafter.

You should always design Your tables carefully, a good design allows you to re-use your tables many times!
While it is possible to use any texteditor(including the Script Editor in MasterPlan) to edit tables, the Table Editor makes it much easier , leaving you to focus on the data itself. The Table Editor makes all the neccesary frequency calculations and makes it easier to group entries together.





10.3.1 Inserting new row

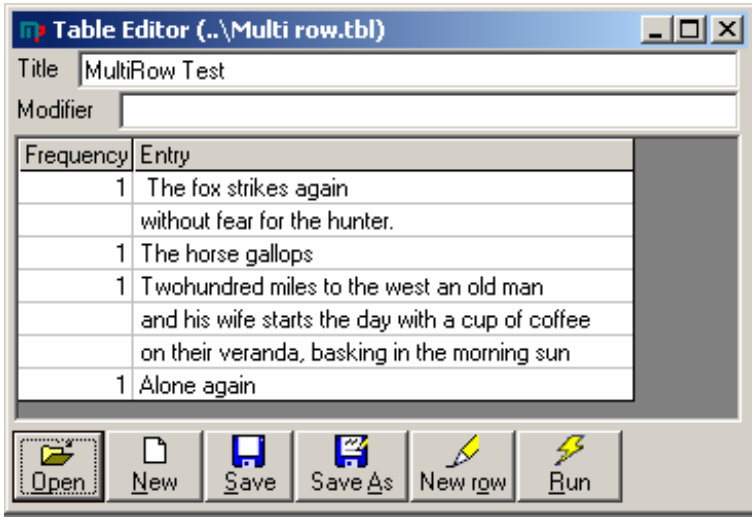
Just select a row and press the Insertkey or use the New Row button.

10.3.2 Deleting row

Just select a row and press the Deletekey. Note: There is NO undo!

10.3.3 Multiple Rows in tables

It is possible for entries to span several rows. Just make sure that the row´s Frequency is blank.



or in code

```
MultiRow Test¤4
1§ The fox strikes again
without fear for the hunter.
2§The horse gallops
3§Twohundred miles to the west an old man
and his wife starts the day with a cup of coffee
on their veranda, basking in the morning
4§Alone again
```

10.3.4 Modifier

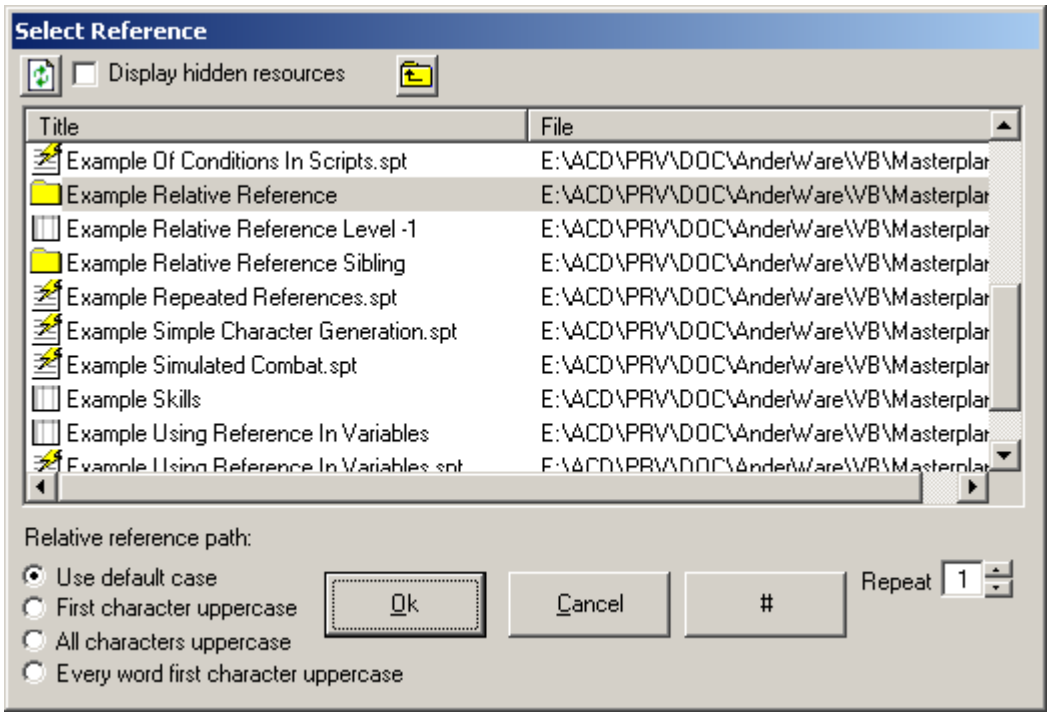
In order to influence the result of a table it is possible to use modifier. This modifier works like normal syntax(see section “Syntax” below)
This modifier will be added to the dice roll made. Most useful with variables.

10.3.5 Clipboard Handling

Cut/Copy/Paste works rowbased on tables. That means that the Table Editor will try to Cut/Copy/Paste entire rows.

10.3.6 Code completion for references

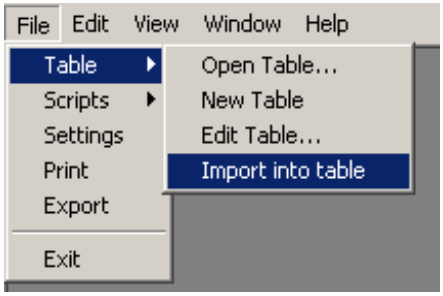
In order to make it easier for the user, code completion is implemented to some extent regarding references. This means that you do´t have to remember exactly every tables filename. Whenever you enter the character for references(#) a list will appear with valid filenames to choose from and their titles as well. Note that this list only displays resources in the same directory as the base script/table.

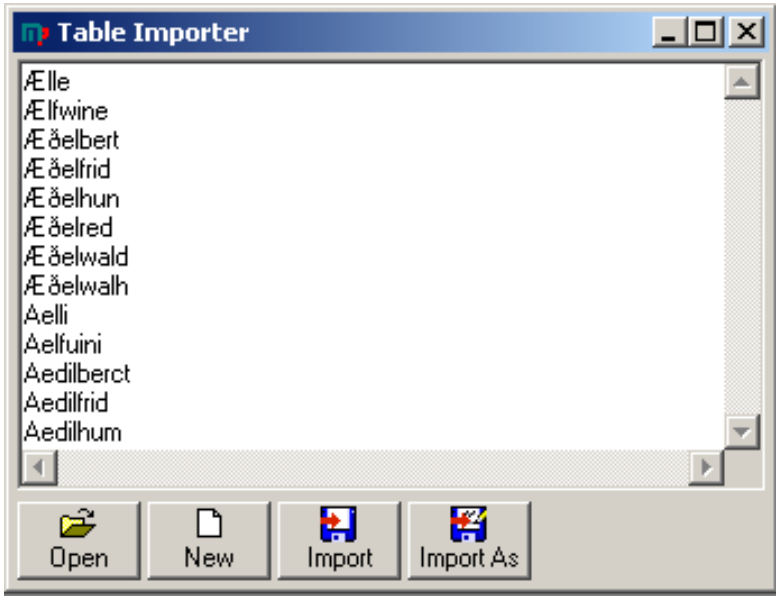


Just select the one you want and press OK. In some cases you might want to enter the reference by hand, like when building advanced relative references. Then simply use the #-button to insert only the #-character. To navigate subfolders, simply doubleclick the folder listed to display the contents of that folder. Use the Up One Level button to display the contnets of the parent folder.

10.4 Import into table

Just get some data in list form, paste in in and violá, instant table!

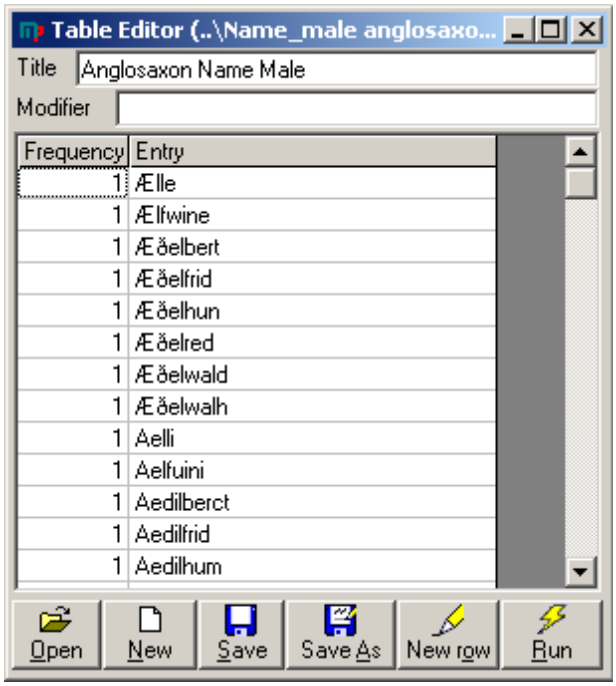




Choose Import

Enter the table´s title and save it

And the result:



10.5 Table Source Format

Though this is automatically handled by the Table Editor, it can be useful to know how it works "behind the scenes".

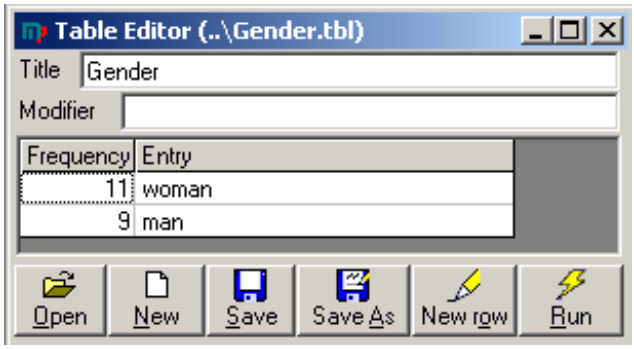
10.5.1 Comments in a table

Comments in the table file are precedeed by semicolon(;). Lines beginning with semicolon are ignored. However, lines containing but not beginning with semicolon will be processed as normal.

10.5.2 Syntax

The first line MUST be Title¤Dicefaces to roll and can also be followed of an VARIABLECHAR that is added to the diceroll.
For more information about available syntax, see below under section Syntax.

10.5.3 Example of simple Table file(*.tbl)

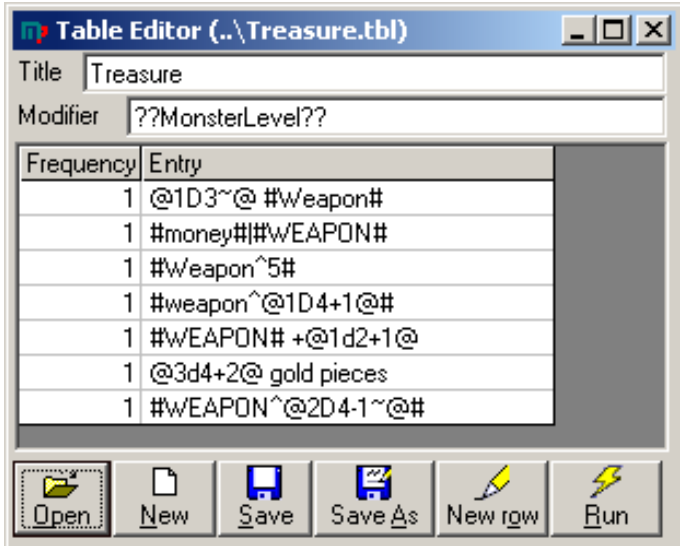


10.5.3.1 Raw file format

```
Gender¤20

11§woman
20§man
```

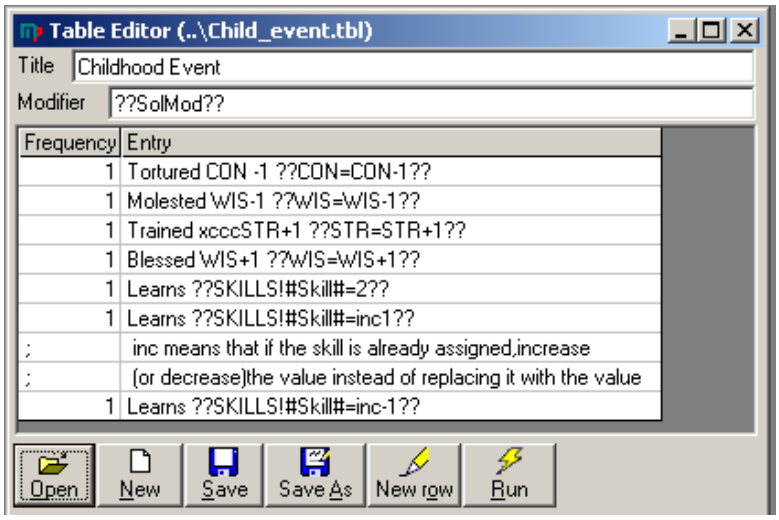
10.5.4 Example of a more advanced Table file(*.tbl)



10.5.4.1 Raw file format

```
Treasure¤6??MonsterLevel??
1§@1D3~@ #Weapon#
2§#money#|#WEAPON#
3§#Weapon^5#
4§#weapon^@1D4+1@#
5§#WEAPON# +@1d2+1@
6§@3d4+2@ gold pieces
7§#WEAPON^@2D4-1~@#
```

10.5.5 Another Table Example.



10.5.5.1 Raw file format

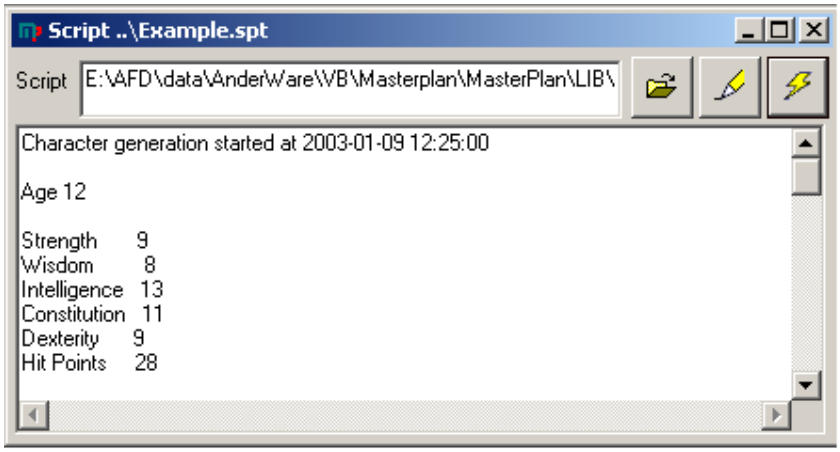
```
Childhood Event¤3??SolMod??
1§Tortured CON -1 ??CON=CON-1??
2§Molested WIS-1 ??WIS=WIS-1??
3§Trained STR+1 ??STR=STR+1??
4§Blessed WIS+1 ??WIS=WIS+1??
5§Learns ??SKILLS!#Skill#=2??
6§Learns ??SKILLS!#Skill#=inc1??
; inc means that if the skill is already assigned,increase
; (or decrease)the value instead of replacing it with the value
7§Learns ??SKILLS!#Skill#=inc-1??
```

11 Scripts

The possibillity to run complex scripts makes MasterPlan powerful. With careful design of scripts and tables it is possible to generate just about anything from a quick NPC to a complete world!

11.1 Running Scripts

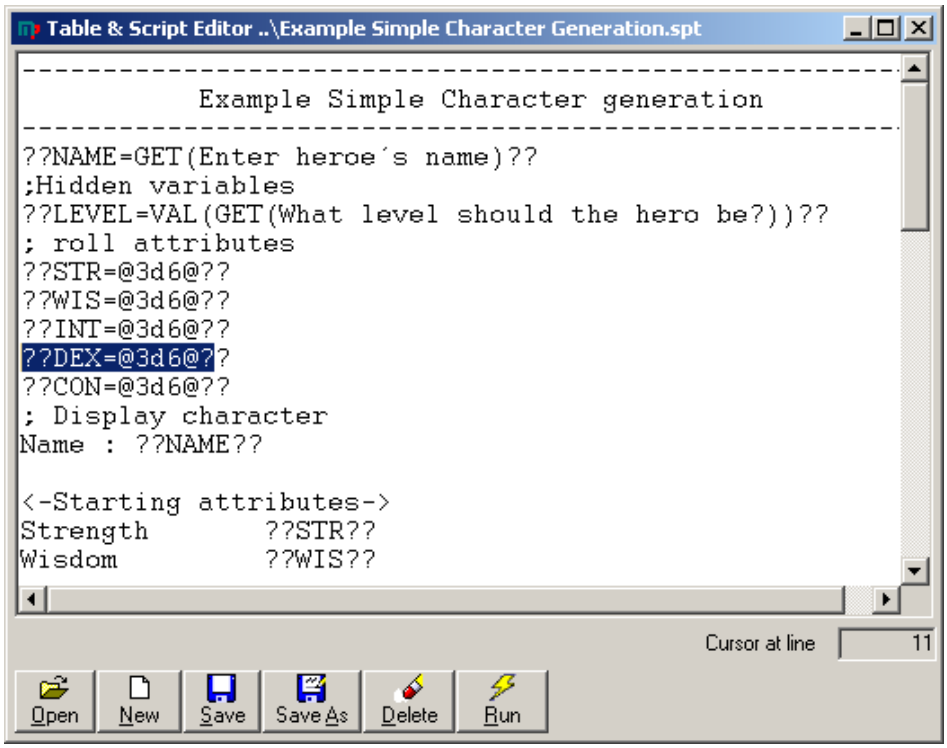
The Open File dialog appears, choose the script you want and wait for it to finish.



To run it again, simply press the Run button. Pressing the 3dot button will open the FileOpen dialog. To edit this script , just press the Edit button.

11.2 Editing Scripts

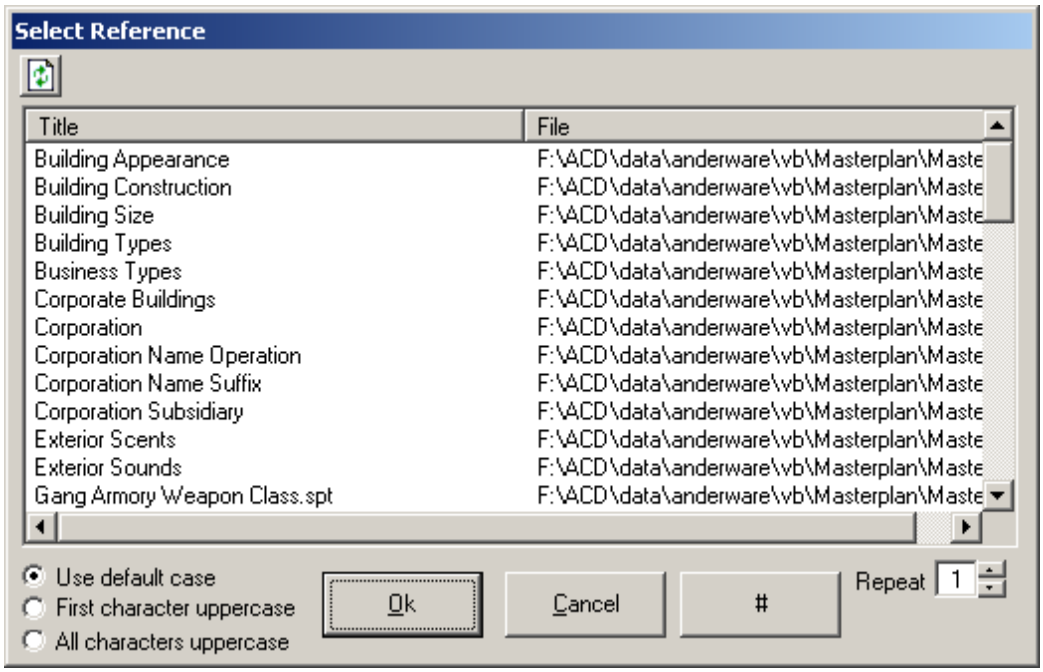
Editing script is just as easy as working in any text editor (like notepad).



The current line that the cursor is positioned at is shown to the bottom right, making it easier to debug scripts.

11.2.1 Code completion for references

In order to make it easier for the user, code completion is implemented to some extent regarding references. This means that you do´t have to remember exactly every tables filename. Whenever you enter the character for references(#) a list will appear with valid filenames to choose from and their titles as well.



Just select the one you want and press OK. In some cases you might want to enter the reference by hand, like when building relative references. Then simply use the #-button to insert only the #-character.

11.3 Script Source Format

The scripts needs more hands-on than the tables, due to their more complex nature.

11.3.1 Conditionals in Scripts

To control the flow through a script it is possible to use conditionals like IF-ELSE.

```
IF  condition AND condition OR condition
statement(s)
ELSE
statement(s)
END IF
```

Note that:
Nesting of IFstatements are NOT supported.
The IF/ELSE/END IF/AND/OR statements are casesensitive, ie must be capitals.
You can not use parenthesis to group AND/OR.

11.3.1.1 Condition Example

```
??Gender=Man??
??x=Man??
*****************************
Test of AND
IF ??Gender??=Man AND ??x??=Man AND 1=1 AND ??Gender??=??x??
  Condition is true
  ??Condition=True??
  Sweeter!
ELSE
  Condition is false
  ??Condition=False??
  Bum!
END IF
*********************************
Condition   ??Condition??
*********************************
Test of OR
IF ??Gender??=Man OR 1=2 OR 1=3
  Condition is true
  ??Condition=True??
  Sweeter!
ELSE
  Condition is false
  ??Condition=False??
  Bum!
END IF
*********************************
Condition   ??Condition??
*********************************
Test of AND  &  OR
IF ??Gender??=Man OR 2=1 AND 1=2
  Condition is true
  ??Condition=True??
  Sweeter!
```

```
ELSE
  Condition is false
  ??Condition=False??
  Bum!
END IF
*********************************
Condition   ??Condition??
```

11.3.2 Logical Jumps in scripts

It is possible to use GOTO statements. The label can be declared before the jump.
Note that the GOTO/LABEL is casesensitive, ie you must use capital letters.

```
statements
GOTO:labelname
statements
LABEL:labelname
statements
```

or

```
statements
LABEL:labelname
statements
GOTO:labelname
statements
```

11.3.2.1 Logical Jump Example

```
Jump test script
??Diceroll=0??
LABEL:Until

??Diceroll=Diceroll+@1D10@??
IF ??Diceroll??<100
??Diceroll??
GOTO:Until
ELSE
GOTO:End
END IF

LABEL:End
Finished
```

11.3.3 Ending Scripts

To terminate a script without running it to its end use END.

```
statements
END:
statements
```

11.3.3.1 Ending Script Example

```
IF ??HitPoints??<0
Target destroyed
END:
END IF
The target still stands
```

11.3.4 Script Example

```
Character generation started at ??NOW??

??NAME=GET(Enter hero´s name)??
;Hidden variables
??LEVEL=VAL(GET(What level should the hero be?))??
??STR=@3d6@??
??WIS=@3d6@??
??INT=@3d6@??
??DEX=@3d6@??
??CON=@3d6@??
??Hit Points=(@1d8@+CON)*LEVEL??
IF ??STR??>16 AND ??CON??>13
??Profession=Fighter??
ELSE
??Profession=Bartender??
END IF
Profession     ??Profession??
Strength       ??STR??
Wisdom         ??WIS??
Intelligence   ??INT??
Constitution   ??CON??
Dexterity      ??DEX??
Hit Points     ??Hit Points??

Sum      ??STR+WIS??
Average  ??(STR+WIS)/2??
Mod avg  ??(STR+WIS)\2??
Round avg ??Rnd((STR+WIS)/2)??
Division ??1/3??
Round ??Rnd(1,75)??
Formula  ??STR+WIS/2??
Sample   ???2*5??
To the power ???2^4??
Trig     ??Cos(180)??
Sqr      ??Sqr(WIS)??
E  ??Exp(1)??
Log ??Log(1)??
Pi       ??PI??
;comment

Equipment
#WEAPON^3#

Character Background
Social Level ??SolMod=@1d2+2@????SolMod??

Childhood Events
#Child_event^4#

Learned skills
??SKILLS!??

Character generation completed at ??NOW??
```

11.3.5 Another script example: Simulated Combat

```
Combat Events
??MikeHitPoints=50??
??SecManHitPoints=25??

LABEL:INI
Initiative
```

```
Mike R:6 = ??M_INI=@1d12~@????M_INI??
SecMan R:5 = ??S_INI=@1d10~@????S_INI??
IF ??M_INI??>??S_INI??
Mike Pistol 6 Attack := ??ATT=@1d12~@????ATT??
??Attacker=Mike??
??Defender=SecMan??
??DEF=6??
??Damage=@1d12~@??
ELSE
SecMan Assaultrifle  5 Attack :??ATT=@2d10~@????ATT??
??Attacker=SecMan??
??Defender=Mike??
??DEF=5??
??Damage=@1d10~@??
END IF
IF ??ATT??>??DEF??
??HITAREA=@1d10@??
??Attacker?? hits ??Defender?? in ??HITAREA??(#hit_area=??HITAREA??#) doing
??Damage?? points of damage!
ELSE
??Attacker?? misses ??Defender??
END IF
IF ??Attacker??=Mike AND ??ATT??>??DEF??
??SecManHitPoints=SecManHitPoints-Damage??
ELSE
??MikeHitPoints=MikeHitPoints-Damage??
END IF

Current
Mike       : ??MikeHitPoints??
SecMan : ??SecManHitPoints??

IF ??MikeHitPoints??<1 OR ??SecManHitPoints??<1
ELSE
GOTO:INI
END IF

IF ??MikeHitPoints??<1
Mike dead, sorry chummer!
ELSE
Rock&Roll chummer you´ve just wasted yet another security guard!
END IF
```

11.3.6 A more advanced script example: Kingdom Creation

```
;Script for MasterPlan
KINGDOMS

;Note that all figures in this script are in km, but can easily be converted to
miles
??UNIT=km??
Total population
;The total population of nations depends on a number of factors, some of which are
discussed in the following sections.
;AREA=total land area
 AREA =??AREA=2500????AREA?? sqr??UNIT??
;WILD=uninhabited land as % of total: 20% is small, 40% is normal, 60% is big
 WILD =??WILD=40????WILD?? %
;WASTE=on inhabited land, land unusable for agriculture, (march, mountains,
desert) as % of total: 10% is small, 20% is normal, 40% is big
 WASTE = ??WASTE=20????WASTE?? %
```

```
;COAST=length of coast: for purposes of food production, each mile (or km) of
coast is assumed to equal 10 squaremiles (or squarekm) of land due to fishing. Of
course, the length of the coastline should be calculated in reasonable scale, say
in 20 mile steps.
 COAST = ??COAST=100????COAST?? sqr??UNIT??
;YIELD=relative yield acquired from the land: 1.0 is normal, 0.5 is poor and 1.5
is great. Yield depends on a number of factors: crop, land quality, agricultural
techniques and climate. Of these, the three first tend to stay relatively constant
over a short time period, but climate may vary a lot from year to year.
 YIELD = ??YIELD=1????YIELD??
;DENSITY=reference population density per squaremile or squarekm in reference
situation: no wasteland, no uninhabited land, yield=1. Density depends on the
subsistance type of the culture. You can also figure the effect of technology and
crop cycles in here.
;       Subsistence       density / sqmil      density / sqkm
;       ---------------------------------------------------
;       Nomad                    5                    2
;       Seminomad               10                    4
;       Semisedentary           50                   20
;       Sedentary              100                   40
;       Industrial             more                 more
;
;       Here are some population densities (actual average densities, _not_
;       reference densities):
;         Artic:                0.3 / sqmil
;         Kalahari desert:      1   / sqmil
;         Canadian praries:     2   / sqmil
;         Medieval England:    30   / sqmil
;         Modern Estonia:      88   / sqmil
;         Modern Hungary:     280   / sqmil
;         Modern Bangladesh: 2400   / sqmil
 DENSITY =??DENSITY=12????DENSITY??

;URBAN = extra people, ie. specialists, leaders and urban population, living from
the surplus of
;the rural population as a percentage. This depends on the margin of living, but
typically
;not directly, as leaders and other specialists will - for reasons that I won't go
deeper
; to - more food (ie. resources) to sustain themselves than rural people. Of
course, their work is
;valuable in increasing the extracted crop yield. Anyway, here are some figures
you can use:
;       Tech level                    urban population
;       ----------------------------------------------
;       Nomadic                              0%
;       Early Farming                        5%
;       Standard Farming                    10%
;       Pre-Industrial Farming              15%
;       Early Industrial                    20%
;       Standard Industrial                 30%
;       Early Technological                 ...
;       Standard Technological              ...
;
;       Here are some estimated RW urban percentages (urban/total population):
;       14th century England:   2% urban
;       14th century Italy:    10% urban (maximum for medieval Europe)
;       17th century Japan:   <20% urban (with intensive rice cultivation)
;       Modern Kenya:          30% urban
;       Modern Argentina:      80% urban
 URBAN=??URBAN=10????URBAN?? %

 POPUL = (AREA+COAST*10) * (1-WILD/100) * (1-WASTE/100) * YIELD * DENSITY *
(1+URBAN/100)
```

```
 POPUL = (??AREA??+??COAST??*10) * (1-??WILD??/100) * (1-??WASTE??/100) *
??YIELD?? * ??DENSITY?? * (1+??URBAN??/100)
 POPUL = ??POPUL=(AREA+COAST*10) * (1-WILD/100) * (1-WASTE/100) * YIELD * DENSITY
* (1+URBAN/100)????POPUL??

 URBANPOPUL=??URBANPOPUL=URBAN/100*POPUL?? ??URBANPOPUL??
Number of cities
 CITIES=??CITIES=rnd(log(URBANPOPUL))?? ??CITIES??
 Biggest city population = ??Rnd(URBANPOPUL/3)??
 Smallest city population=??Rnd((URBANPOPUL/1,5-(URBANPOPUL/3))/(CITIES-1))??


Military Power

;TL= Technological Level where TL(military) and TL(militia) are got from the table
below:
;    TL                            military  militia Marine
;    ---------------------------------------------
;    Always nomadic                  0         2,0    0
;    Occasionally nomadic            0,5       1,5    0,5
;    Farming                         1,0       1,0    0,75
;    Trading                        1,5       0,75   1,0
 TL_MILITARY=??TL_MILITARY=1????TL_MILITARY??
 TL_MILITIA=??TL_MILITIA=1????TL_MILITARY??
 TL_MARINE=??TL_MARINE=0,9????TL_MARINE??
;MOBILISATION=mobilisation level of the kingdom
;     Kingdom attitude
;     None       0
;     Peaceful        5
;     Normal         10
;     Aggresive     15
;     Killing Spree 20

 MOBILISATION=??MOBILISATION=10????MOBILISATION??

;MILITARYFACTOR = the percent of military personnel of population (default =
MOBILISATION * TL(military))
;MILITIAFACTOR = the percent of possible militia in a population (default =
MOBILISATION*2 * TL(militia))
 MILITARYFACTOR = ??MILITARYFACTOR=MOBILISATION*TL_MILITARY??
??MILITARYFACTOR??
 MILITIAFACTOR = ??MILITIAFACTOR=MOBILISATION*2*TL_MILITIA??    ??MILITIAFACTOR??
 MARINEFACTOR = ??MARINEFACTOR=MOBILISATION*TL_MARINE??    ??MARINEFACTOR??

;To get the size of army simple calculate:

 MILITARY =??MILITARY=POPUL*MILITARYFACTOR/100??  ??MILITARY??
 MILITIA  =??MILITIA=(POPUL-MILITARY)*MILITIAFACTOR/100??    ??MILITIA??

The military is the number of full-time soldiers available for war.
The militia is the number of part-time militia that can be raised in times of
war.|
;To get the size of navy, try this:
 NAVALSHIPS=(COAST/(AREA+COAST)) * MILITARY/60 * MARINEFACTOR
 NAVALSHIPS=(??COAST??/(??AREA??+??COAST??)) * ??MILITARY??/150 * ??MARINEFACTOR??
 NAVALSHIPS = ??NAVALSHIPS=Rnd((COAST/(AREA+COAST)) * MILITARY/150 *
MARINEFACTOR)?? ??NAVALSHIPS??

 HEAVYSHIPS = NAVALSHIPS / 100 * MOBILISATION
 HEAVYSHIPS =??HEAVYSHIPS=Rnd(NAVALSHIPS/100 * MOBILISATION)?? ??HEAVYSHIPS??

 SAILORS = TL_MARINE*10 * (NAVALSHIPS+2*HEAVYSHIPS) * MARINEFACTOR
 SAILORS =??SAILORS=TL_MARINE*10 * (NAVALSHIPS+2*HEAVYSHIPS) * MARINEFACTOR??
??SAILORS??
```

```
Production:
;       { POPUL*2 / 10000,                    if TL is (occationally) nomadic
;GNI = {
;       { URBANPOPUL*5+(POPUL-URBANPOPUL)*3)/10000   otherwise
 ??GNI=(URBANPOPUL*5+(POPUL-URBANPOPUL)*3)/10000??
 GNI=(URBANPOPUL*5+(POPUL-URBANPOPUL)*3)/10000= ??GNI??

Power components:
 ??ECONOMICPOWER=GNI/5??
 ECONOMICPOWER =GNI/5 = ??ECONOMICPOWER??
 ??MILITARYPOWER=((MILITARY+SAILORS)/1000) + (MILITIA/3000) + ((NAVALSHIPS +
3*HEAVYSHIPS)/10)??
 MILITARYPOWER =((MILITARY+SAILORS)/1000) + (MILITIA/3000) + (NAVALSHIPS +
3*HEAVYSHIPS)/10) = ??MILITARYPOWER??
 ??POPULATIONPOWER=POPUL/100000??
 POPULATIONPOWER =POPUL/100000 = ??POPULATIONPOWER??

 Total power:
 ??POWER=ECONOMICPOWER + MILITARYPOWER + POPULATIONPOWER??
 POWER = ECONOMICPOWER + MILITARYPOWER + POPULATIONPOWER = ??POWER??

;Originally Compiled by Juha.Vesanto@hut.fi
;http://www.hut.fi/~vesanto/
;This page is largely based on discussions on various mailing lists, in particular
the
;world-design mailing list / Agricultural land area-thread. People who wrote to
that thread
;will find whole paragraphs of their text below practically untouched, without
quotation marks.
;Sorry about that, but I really saw no reason not to use perfectly good text as it
was.
;Thanks to the following people for giving their input to the discussion (in no
particular order): Hartley Patterson,
;Mike Harvey, Colin Watson, Thomas Hudson, Brett Evill, Chris Lea, Mike Schmitz,
Ken St-Cyr, Chris Tutty,
;ANDOVER@delphi.com, Silburn Luke, Burton Choinski, Lee W. Dowd, Andrew B. Watt,
Mike Maxwell A Lapalme,
;scott david orr, Rob Dean, Kevin Rose, Ian Smith, Michael Bege

;adapted and tweaked to MasterPlan standards by Ramses
```

12 Syntax

Tables and scripts shares the same syntax.

12.1 Control characters used in scripts and tables

Control character	Description	Example
¤	separates title from number of entries	`Treasure¤6`
§	paragraph start	`1§Gold`
#	reference Case sensitive referencing #NAME# returns “ANDERS FORSLUND”, if the first TWO characters are upper case. Converts entire expression. #Name# returns “Anders forslund”, if the first character is uppercase but not the second. Converts the first character of the expression and leaves the rest as it is #name# returns the actual content from the resource file , if both the first two character is lowercase. Converts nothing, returns “anders forslund” Uppercase first letter of each word #NameE# returns “Anders Forslund” Relative referencing(like HTML) The relative referencing supports multi level referencing, i e the relative referenced resource can reference another resource. There is no theoretical limit to how many levels referencing should work but at some point your computer may run out of memory returning a “Out of stack space”-message. This will probably not happen unless you make a circular reference as a misstake. #..\Name\Fantasy_name# - one level up and into an other folder down. With this you can easily re-use your general tables like "Color" or "Direction" e t c by putting them in a General folder next to the Fantasy folder for instance. #..\Fantasy_name# - one level up #..\..\Fantasy_name# - two levels up. Actually you can have several “..\” groups, each group will go one level up. #NAME\Fantasy_name# - one level down #City\Events\Religious_event”# - means two levels down. Each “\” goes one level up. Reference Scripts is also possible, this makes the naming of tables and script important, don´t use the same name for both a table and a script unless you know what you are doing. The software gives precedence to tables, ie if both a script and a table have the same name, the reference will use the table.	`#Weapon#`
;	comment	`;Created by Ramses`

@	invokes diceroller	@1d2+2@
^	repeats the reference with each returned entry on a new line #WEAPON^3# could return Axe Sword Spear	#WEAPON^4#
^_	repeats the reference with each returned entry on the same line #WEAPON^_3# could return AxeSwordSpear Notice there is no space between the items. Useful for generating names, languages etc.	#WEAPON^_4#
=	Sets value for lookup reference, disregards dieroll and modifier completly. #WEAPON=3# will get the entry in WEAPON as if rolled 3. Setting this lookup to a higher value than entries existing in the table returns the highest entry.	#WEAPON=3#
^,=	Repeat has preference, #WEAPON^4=3# will get four(4) entries(identical) in WEAPON as if rolled 3	#WEAPON^3=3#
|	inserts a new line	|
~	dice rolls are openended, meaning that whenever a die rolls its highest number, that die is rolled again add added to the sum.	§@1D3~@
??	variable or expression. You can group your variables like "ATTRIBUTES!STR" where the part before the exclamation mark(!) is the groupname. This can be useful in table you have no way of knowing what they will return. To get the complete listing of a group use ??ATTRIBUTES?? Referencing a non-assigned invisible variable returns 0 Variables can be used everywhere. ??variable_name=1?? Assigns the variable “variable_name” the value of 1 Variables can be assign both numerical and alphanumerical values. **Reserved Variable Names is used by the software in order to extend special functions for advanced scripting** DUPLICATESACCEPTED, the number of duplicate entries that are required before a script/table accepts a duplicate entry generated. Default value is 2 meaning that if a script will not accept the same generated entry directly after the the first. If set to 10 the script will try to get another result 10 times before giving up and accepting the duplicate. If set to 0, no duplicate entries are accepted. Varning, do not use this unless you know what you are doing, it is possible for the script to execute an infinite loop (although you can break the loop with the “Abort Running Processes”-command, se above). It is possible to use several different values during the execution of a script.	??Strength?? ??1+2?? ??Strength=@3d6@?? #Skill^??Level??# #HitArea=??Damage??# ??ATTRIBUTES!STR=15?? ??ATTRIBUTES!INT=12?? ??ATTRIBUTES!WIS=16?? ??ATTRIBUTES?? ??DUPLICATESACCEPTED=10??

12.2 Keywords

Mathematical expressions	Description	Example	Returns
+,-,*,/	Normal arithmetic	??1+1??	2
\	Integer division	??8 \ 3??	2
^	Exponent	???^4??	16
%	Mod. Returns the remainder of a division	??8 % 3??	2
<,>	Lower than/greater than		
<=,>=	Lower than or equal /Greater than or equal Note that the equal sign must be to the right of the size sign.	IF 1<=2	TRUE
Mathematical functions	**Description**	**Example**	**Returns**
Cos,Sin,Tan	Trigonometric functions	??Cos(180)??	-1
Sqr	Square	??Sqr(2)??	4
Exp	antilogarithm. returns e raised to a power	??Exp(1)??	2,71828182845905
Log	logarithm, Returns the natural logarithm of a number	??Log(1)??	0
Constants	**Description**	**Example**	**Returns**
PI	Natural Pi number	??PI??	3,14159265358979
Now	constant(returns current system date and time))	??Now??	2002-08-13 08:37:53
Other functions	**Description**	**Example**	**Returns**
Get(string)	gets user input, string is the text that will be displayed.	??NAME=GET(Enter heroe´s name)??	Whatever the user entered
Rnd	Round	??Rnd(1,75)??	2
Val	converts strings to numbers	??LEVEL=VAL(GET(What level should the hero be?))??	Makes sure that the variable LEVEL will be a number just in case the user entered a nonnumeric

12.3 Reserved keywords in scripts

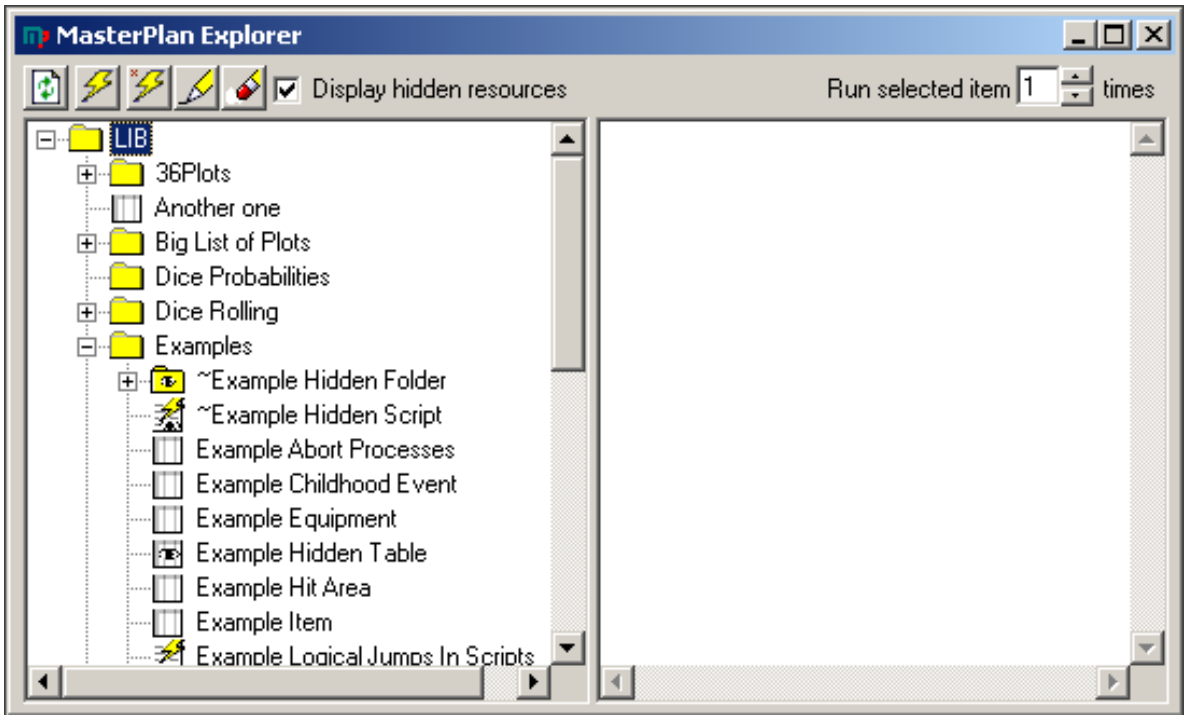
IF,ELSE,END IF, END:, GOTO:,LABEL:

12.4 Number handling

Maximum whole number in mathematical operations is 999999999999999. Numbers larger than this will be presented in scentific notation (1E+15). The data type used holds double-precision floating-point numbers as 64-bit numbers in the range -1.79769313486231E308 to -4.94065645841247E-324 for negative values; 4.94065645841247E-324 to 1.79769313486232E308 for positive values.

12.5 Hidden resoucrce files/folders

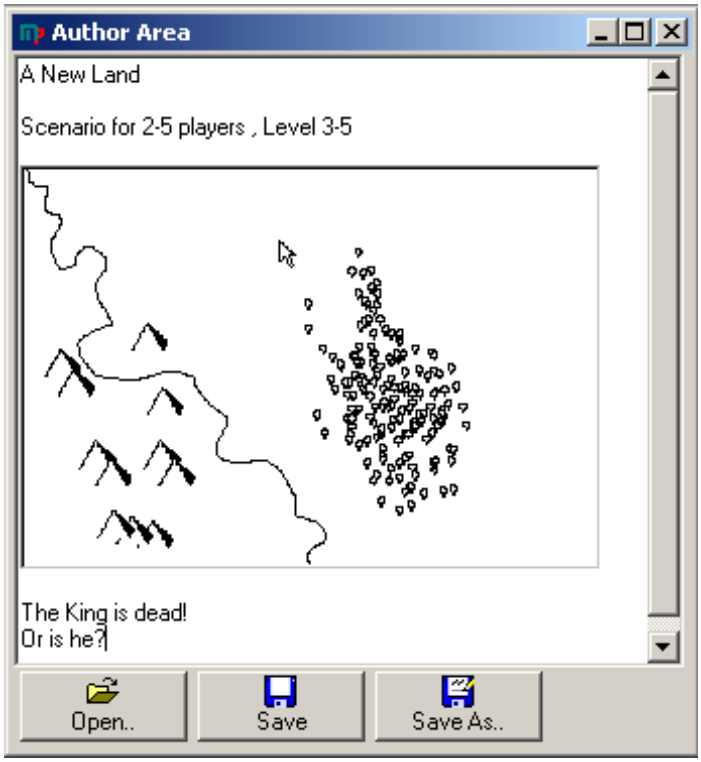
Files(tables and scripts) and folders beginning with a "~"-character is treated as hidden files and will not be displayed in the MasterPlan Explorer tree if not the "Display Hidden Files" checkbox checked.



Hidden resources will be shown with a special icon. These resources are accessible just as normal files, the only difference is that they are not displayed in the MasterPlan Explorer.

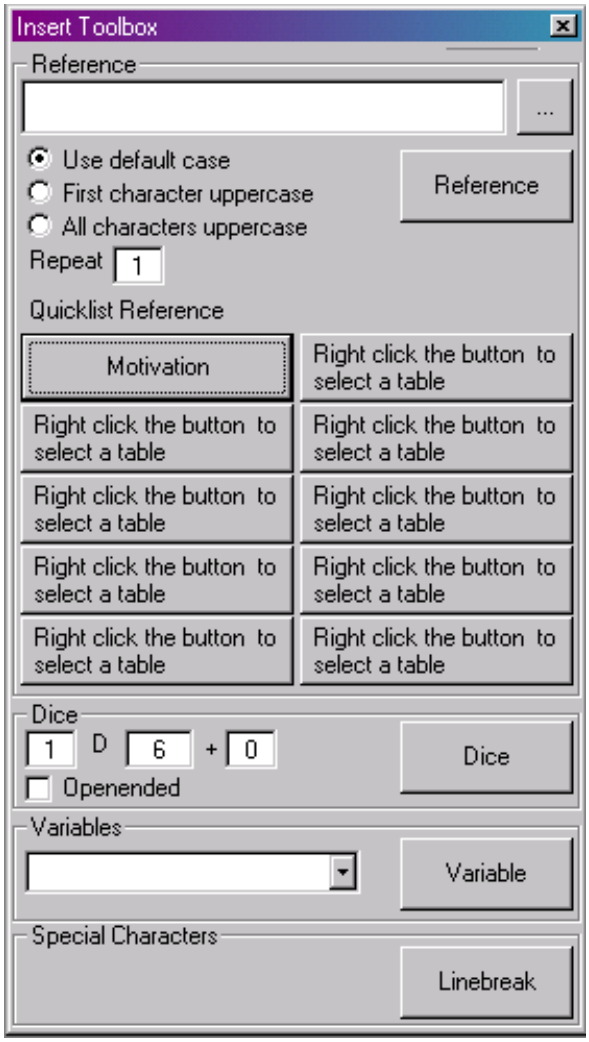
13 Author Area

Working text area - For the struggling author. A place to keep notes,comments and any text. An entire scenario/adventure could be written here. Saves as RichText(*.RTF) with picture support. Use Cut/Copy/Paste to enter pictures.



14 Insert Toolbox

This tool makes it even easier to create custom scripts and tables.

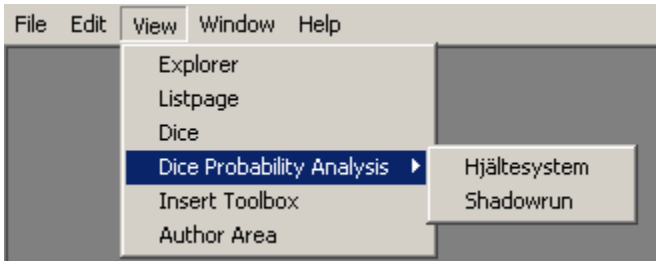


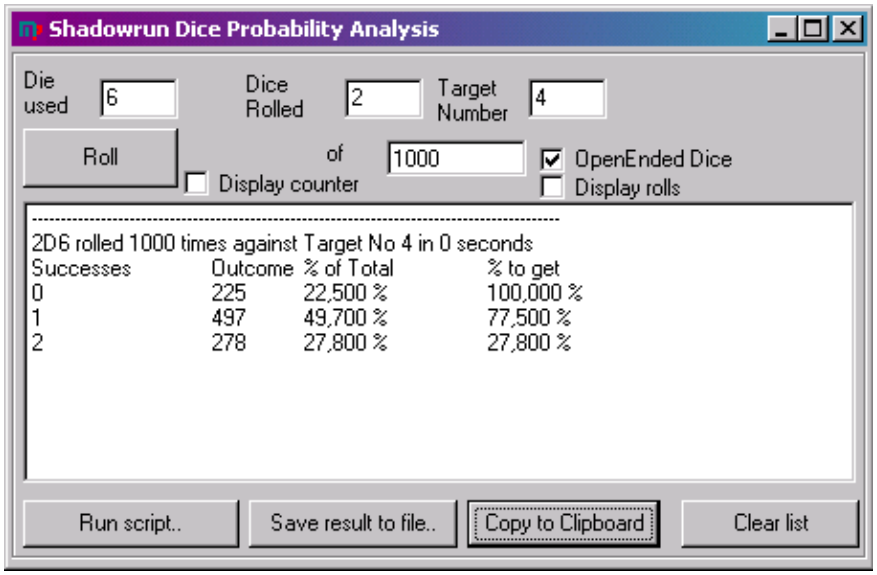
Simply click the Reference, Dice, Variable or Linebreak button to insert it into the script you were working on at the position the cursor were.
Not fully compatible with the Table Editor. Yet.

15 Dice Probablility Analysis

Some have game mechanics that are quite difficult to calculate probablilities, therefor the probablility analysis modules.
These modules were specifically developed for Hjältessystem and Shadowrun
but can be used for other game systems as well. To get reasonably good statistically results it is neccesary to roll perhaps a million time or more and this of course takes a while. To make things simpler, is it possible to use scripts in
the Dice Probabilility Modules to enable batch operations.

15.1 Shadowrun





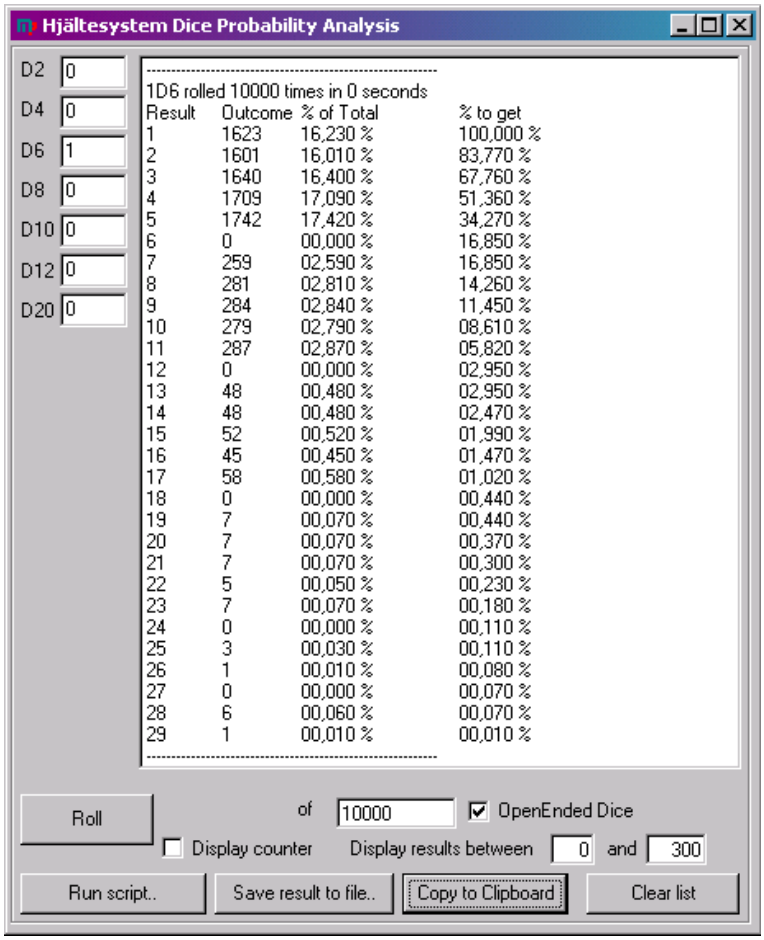
Just enter die used(6-sided default), number of dice to roll, target number, check the OpenEnded-box and how many times we should roll it. Then click Roll, sit back and wait for the software to finish. Pressing the Run Script button opens the Open File dialog where you can choose a batch script file.

15.1.1 Example of script to Dice Probability Module(Shadowrun)

```
[Autorun]
OpenEnded=1
Rolls=10000
DieFaces=6
[Values]
Dice1=3
Dice2=4
TargetNumber1=3
TargetNumber2=4
TargetNumber3=5
```

and so on upto any(almost) number of dice and target number possible.

15.2 Hjältesystem



15.2.1 Example of script to Dice Probability Module(Hjältesystem)

```
[Autorun]
OpenEnded=1
Rolls=10000
[Values]
V1=T2
V2=T4
V3=T6
V4=T8
V5=T10
V6=T12
V7=T10+T4
V8=T10+T6
V9=T10+T8
V10=T10+T10
V11=T10+T12
V12=T20+T4
V13=T20+T6
V14=T20+T8
V15=T20+T10
V16=T20+T12
V17=T20+T10+T4
V18=T20+T10+T6
```

```
V19=T20+T10+T8
V20=T20+T10+T10
```
and so on