



IMMORTAL CITIES:
CHILDREN OF THE NILE

SCENARIO AND CAMPAIGN EDITORS' MANUAL

BY
YAHYA

Table of Contents

SCENARIO EDITOR

Table of Contents.....	2
Acknowledgements	4
Notes to Scenario and Campaign Editors.....	4
Scenario Editor	5
Starting the Scenario Editor.....	5
Loading, Saving, and Testing Scenarios.....	5
Map Page.....	6
Elevation Files	6
Elevation Page.....	8
Land Elevation	9
River Creation	10
Terrain Page.....	10
Terrain	10
Remove Sprites.....	11
Paint Roads.....	11
Units Page	11
Other Unit Controls	12
Triggers Page.....	12
Component Names and "Autonames"	14
Trigger	14
Condition.....	17
Effect.....	19
Object.....	26
Area.....	29
Players Page.....	30
Events	30
Worship.....	30
Construction.....	30
Options.....	30
Story Page.....	31
Debrief Page.....	32
Goals Page	32
Goal List.....	32
Name	32
Effect of Goal.....	32
Goal Category	33
Player Attribute for Goal.....	33

Qualifier	33
Amount.....	33
Delay	33
Duration in Seconds	33
Goal Overview.....	33
Scenario Tips	33
Reset Some Entry Fields.....	33
Delete Goal.....	34
Replace Goal.....	34
Submit as Goal.....	34
World Level Page	34
World Map Selection.....	34
New Scenario Creation	35
New Activation Site.....	36
Starting the Campaign Editor.....	39
Using the Campaign Editor.....	39
Background Image	39
Campaign Name	39
Campaign Description (Optional).....	40
Campaign Tier.....	40
Campaign Tier Description.....	40
Scenarios in Campaign.....	40
Add Button.....	40
Remove Button.....	41
Update Button.....	41
Scenario Information	41
Available Scenarios.....	42
New Button.....	42
Load Button.....	42
Save Button.....	42
Exit Button.....	42
Appendix 1: Trigger Example.....	43
Example	43
Appendix 2: Scenario Optimization	45
Possible Performance Hits	45
Creating Smarter Triggers	46
Index.....	49

Acknowledgements

This manual has been compiled using information as contained in the original "Immortal Cities: Children of the Nile" (abbreviated as "CotN" throughout) manual, written by Ken_P and published by Tilted Mill, dated October 22, 2004.

It has been further flushed out using information from the absolutely excellent "Empires: Dawn of the Modern World Scenario Editors (sic) Manual" (referred to throughout as "Empires: DMW SEM"), written by David Boghdan and published by Stainless Steel Studies, dated circa August, 2003. (Please note: this date is approximate, based on the print date of my copy of the Empires: DMW SEM of August 1, 2003. I have not verified the actual publication date of the manual, as I deemed it unnecessary.)

Using the framework of the Empires: DMW SEM, and conducting tests on every item in the CotN Scenario Editor, this manual was meticulously assembled. I'm sure errors will be found. Please contact [Yahya](#) at the CotN forums to inform me of any you find, and I will issue a revision if the error is confirmed. The point of this manual is to make it as detailed and accurate as possible.

I also must acknowledge the work of [Raccoon_TOE](#) who has helped me time and again. Without his work in the [Custom World Maps](#) and [Elevation Maps](#) threads of the CotN forums, neither this guide nor many of my own scenarios would have been possible. As such, I thank him immensely.

I thank [Karter](#) for his help with camera scripting in the [Camera Tutorial](#) thread at the CotN forums. Between his advice there and his work on [The Keep](#), I learned a considerable amount about scripting cameras, not to mention the Triggers Page, that made the largest and most detailed section of this guide possible.

Lastly, I thank the great community of people at the CotN forums, for whom this guide was made. With so many people who both enjoy building and playing CotN scenarios, we've all had a great time together in the past five-plus years, and I see no end in sight.

Notes to Scenario and Campaign Editors

CotN includes both a Scenario Editor and a Campaign Editor so that one can design and create maps, scenarios, and campaigns. These editors are the same powerful tools Tilted Mill used to create the single player campaigns that shipped with the game. This manual provides a guide and reference to help scenario and campaign editors learn as much about the editor as possible.

Also, please note that to create a Campaign, one must first create all the scenarios which will be included in it. A Campaign adds in only the completed scenarios, and is therefore the last section of this manual.

It is highly recommended that Scenarios and Campaigns created for CotN be thoroughly tested if they are to be shared with others. If possible, test work on more than one computer. Bear in mind a wide variety of computer systems exist, and a scenario that runs well on one computer may not run well on another if, for example, it has less RAM or an older video card. See the Appendices for tips on optimizing scenarios.

This manual was written while conducting research and tests using CotN version 2.1.0.6 (Alexandria). Not all information contained herein is applicable to CotN version 1.3.0.1 (CotN Enhanced Edition). As such, where information is applicable only to the Alexandria version, it is so noted by including "Alexandria only" in parenthesis.

Scenario Editor

The CotN Scenario Editor provides all the tools needed to create original scenarios. Designers can build their own CotN maps, add triggers to control victory conditions or in-game events, set game and player conditions, and include instructions, dialogue, or other messages. Please note that, because the Scenario Editor is so large and versatile, it cannot be described in its entirety in this manual. Trying different things and experimenting with the multitude of options are the best ways to become familiar with the power and flexibility of the Scenario Editor.

The Scenario Editor is separated into ten main sections or “pages.” Icon buttons located along the bottom-left of the screen take you to each page. The buttons on the bottom-right of the screen provide save, load, and testing options, as well as the means to exit the editor. Each of these options will be discussed in this manual in turn.

TIP: Right-click and drag does not change the view angle in the editor as it does in the game. To see the game world from different angles while in the Scenario Editor, hold the ` key (grave, which is the tilde [~] key without pressing Shift) on your keyboard. Press the Home key to return to the default view.

Starting the Scenario Editor

To start the Scenario Editor, click on the “Editor” button on the Main Menu of CotN. In the Editor menu, click on either the Create Scenarios button to make a new scenario from scratch, or the Edit Scenarios button to modify an existing scenario. Either button will open the Scenario Editor, but the latter will allow selection from the existing scenarios in the Scenarios folder of the applicable installation (1.3.0.1 or Alexandria).

Loading, Saving, and Testing Scenarios

The buttons on the lower-right of the Scenario Editor, just left of the Mini-map, provide access to the Scenario Editor’s various save and load options. From left to right and top to bottom, these buttons are:

- **Quick Load** - Loads up the most recent Quick Saved scenario into the Scenario Editor. This will only work if a scenario has been previously Quick Saved.
Hot Key for Quick Load: / (forward slash key)
- **Save** - Opens up the Save screen so a designer can name a scenario or overwrite previously saved files. Scenarios saved using the Save screen are located by default in the ../Data/Scenarios directory of CotN Enhanced Edition, and the My Documents/Tilted Mill/Children of the Nile/Scenarios directory of CotN Alexandria Edition.
- **Test** - Launches CotN and loads the scenario currently being edited in the Scenario Editor. When testing, the editor may be reentered by clicking the “Back to Editor” button on the Game Options menu.

NOTE: It is highly recommended that scenarios not be tested in this way, as unpredictable results can occur, including permanently and irreparably bugging the scenario. Rather, save the scenario and exit the Editor, and play the scenario normally by selecting it from the Main Menu.

- **Quick Save** - Saves the scenario currently being edited to QuickSaveScen YYYY.MM.DD - HHh.MMm.SSs.scn in the applicable directory as noted above. Each Quick Save creates a new Quick Save file, and the latest date is the one loaded when Quick Loading.

- **Load** - Opens the Load screen so a designer can pick a previously saved scenario to open in the Scenario Editor.
- **Exit** - Returns the designer to the Editor menu.
- **Tutorial Mode Checkbox** - To the left of Quick Load button is a checkbox with no label. After many years of wondering, I discovered what this button does. It enables what I will call "Tutorial Mode." When checked, the Load button opens the Tutorial folder(s) of the CotN installation, and the Save button will save the scenario to the same folder(s) with a .TUT extension. If unchecked, the scenario builder uses the default Scenarios folder, and saves the scenario with a .SCN extension.

Map Page

The first thing to do when creating a scenario is to create a map. This is the nome level: the map on which the player will build her city. The default map is a 100 x 100 map covered with a grass texture, but a different map can be generated if desired. There are numerous options from which to choose.

To create a new map:

1. Set the Map Size. A Map Width and Map Height must be entered, measured in grid tiles. For example: 100 x 100 or 60 x 30. (Widths and heights from 15 to 400 are possible.)

NOTE: Larger maps will be far more demanding on slower-performing computers.

2. Select a Default Terrain. Nearly any of the game's terrain textures can be chosen as the default, however some textures will create non-buildable terrain or might cause an extreme slowdown in computer performance, even with no other items on the map.
3. To generate the new map, click the "Create new scenario" button.

NOTE: Pressing this button will erase everything in the scenario, including all triggers and text. Hence, map creation will be the prudent scenario designer's first step.

Also contained in the top right of the Map page is a display of the current camera control setting in the editor. By pressing F2, the designer may select either Normal or Full camera control. Each of these options performs identically to its counterpart in-game.

Elevation Files

Advanced users can create an Elevation File to produce varied topography from real-world elevation data. This requires some work ahead of time, but the results can be spectacular.

The Elevation File is used in conjunction with generating a Blank map. To make use of this feature, elevation data must first be downloaded from the Internet. A good place to look is a web site such as the National Geophysical Data Center (NGDC) of the National Oceanic and Atmospheric Administration (NOAA), U.S. Department of Commerce.

Here are two example sites that can be used to create elevation files:

<http://www.ngdc.noaa.gov/cgi-bin/mgg/ff/nph-newform.pl/mgg/topo/customdatacd> (Globe: Select your own area)

OR

http://www.ngdc.noaa.gov/mgg/gdas/gd_designagrid.html (GEODAS Grid Translator).

My personal favorite is the Globe database, as I find it produces better results. The GEODAS Grid Translator occasionally produces an .flt file rather than a .bin file, making the data unusable. I have yet to find a solution to that problem, so I use the Globe database.

This process is much simpler than it might sound. Here is a step-by-step example:

1. Choose your area. For this example, we will choose the U.S. state of Rhode Island, along the Narragansett Bay, because it's my book. However, any area of the world may be chosen at the reader's discretion.
2. On the Globe page, one can either manually enter coordinates by selecting the radio button "Text-Entry," or select an area on a JavaScript map by choosing "Map-Based." The Java map is rather small, however, so I find it best to use Google Earth to find the right coordinates and then enter them manually. The coordinates must be entered in degrees and decimals, rather than minutes and seconds, and North latitudes and East longitudes are positive. Also, it is important, if one wishes to be accurate, to select a 1:1 ratio of Earth area. Otherwise, there will be black area around the nome in game. For this example, I chose an area of 40.0 degrees North to 41.0 degrees North by 70.9 degrees West to 71.9 degrees West. So this is 41.0, -71.9, -70.9, 40.0.
3. Use all the default options below the coordinate entry, though it is helpful to change the File name, especially if making several elevation files. I chose "NBOAs" as the file name, for Navy nostalgia.
4. Click "Get Data." A new screen will appear with three files: NBOAs.bin, NBOAs.hdr, and NBOAs.fmt. We only care about the first two.
5. For the next step, a folder named "Elevation Maps" must exist under the CotN installation at ../data/scenarios/ (Either the 1.3.0.1 or Alexandria installation path). Make sure this is created before continuing.
6. Right-click NBOAs.bin and click "Save As...". A box will appear requesting the location. First, change the file extension from .bin to .dat. (This can be done later in Windows Explorer, but this is just as easy.) Navigate to the "Elevation Maps" folder, either for 1.3.0.1 or Alexandria, and save the file there.
7. Right-click NBOAs.hdr and click "Save As...". The same box will appear, but the location should be correct this time. If not, navigate there again. Change the file extension from .hdr to .txt and save the file.
8. Now open Windows explorer and navigate to the "Elevation Maps" folder. Open NBOAs.txt (it should open in Notepad) and note the figures for "number of rows" and "number of columns." In this case it is 90 and 120, respectively. Select everything in that file, and type the column number first, a space, and the row number. In this case it should be "120 90". Save the file and close it.
9. If the Editor is already running, it must first be exited and reloaded to allow selection of this elevation file.
10. Start the Scenario Editor and choose the Elevation File from the drop down list on the Map page. I chose NBOAs. Once this is selected, some controls disappear while new ones

appear. Only a width must be chosen for the map (I recommend choosing 250 for this example to see the effects) because the height will be automatically scaled to the same ratio as the width and height listed in the .txt file from step 8.

The new controls that appear are as follows:

- **Water Depth** - This is the maximum depth the water will be, Measured in CotN elevation. I chose 4.
 - **Water Cutoff** - This is the depth, in meters, that is set to be equal to the Water Depth value. If there are depths greater than this number in the data, they will be cut off at (that is, set equal to) the maximum water depth. For this map, I played around with it before settling at a figure of 50 meters.
 - **Land Height** - This is the maximum height that land will reach in CotN elevation. This can be played around with as well. I set it to 20 in this example. Changing this figure as well as the next will have a dramatic effect on the land contour.
 - **Land Cutoff** - This is the height, in meters, that is set to be equal to the Land Height value. If there are heights greater than this value in the data, they will be cut off at (that is, set equal to) the maximum land height. I set this to 1500 for this example, to give a nice, smooth area to build on.
11. Once Elevation File has been chosen, the Map Width and the Water and Land values set, click the Create Map button to generate the new map. Once this is done, a nice map of Rhode Island's Narragansett Bay, as well as a good portion of Long Island Sound, should be produced on your screen. Have a look at the mini map in the lower right corner, and compare it to Google Earth to see the effects. I chose a really large region to create a recognizable map. I'd never do this for a scenario.

NOTE: Creating a map in this way will create some rather odd textures, specifically some purple areas on the map. That's all right however, because they will be painted over during the Terrain phase anyway. Also, using Elevation Maps to create realistic maps of the Nile River Valley is impractical because there is no depth information given for rivers. However, a valley can be an indication where the river should be carved, and using an Elevation Map can save the designer hours of design time in making a new scenario.

Elevation Page

Use this page to vary the map's topography by creating hills, cliffs, valleys, and bodies of water. There are 65 elevation levels for the land, from -24 to +40, and the Fine Elevation allows decimal adjustments, from .00 to .99 (0 to 99 on the slider).

The default terrain level is approximately 4.90 (4 Terrain Elevation, 90 Fine Elevation). Land that is set to an elevation of less than -0.1 (-1 Terrain Elevation, 99 Fine Elevation) is under water during a normal flood level. Any land below 0.1 (0 Terrain Elevation, 1 Fine Elevation) is under water during a High Flood level. Finally, any land below -1.11 (-2 Terrain Elevation, 89 Fine Elevation) is under water at all times. (See **River Creation** later in this section to learn about the CotN river.)

NOTE: An elevation of -1.51 (-2 Terrain Elevation, 49 Fine Elevation) or below is impassable by CotNizens, and they will build a small reed boat to cross the body of water if there is no other way around. Any elevation of -1.50 (-2 Terrain Elevation, 50 Fine Elevation) or above is considered "shallows" and is passable; CotNizens will merely wade through the water.

The Slope Elevation slider controls how far the terrain brush radiates out to the surrounding tiles. At a Slope Elevation of 100, only the tiles directly under and adjacent to the brush are affected. At a Slope Elevation of 0, the entire map is set to the chosen height.

Land Elevation

To lower or raise the land:

1. Be sure the Elevation page is selected.
2. Select a Brush Size with which to “paint” the chosen elevation. The options are:
 - a. **Tiny:** a 1x1, or 1 grid square area
 - b. **Small:** a 3x3 grid area
 - c. **Normal:** a 5x5 grid area
 - d. **Large:** a 7x7 grid area
 - e. **Huge:** a 9x9 grid area
 - f. **Embiggened:** an 11x11 grid area
3. Set the Terrain Elevation slider to the desired elevation level. Remember that negative elevations are under water, and the specific settings chosen below 0 determine where the river will be at Normal and Low Flood. The slider can be clicked above or below the “knob” to increment or decrement by 1.
4. Select either Paint Hills, Paint Cliffs, or Paint Cliff Paths:
 - **Paint Hills** - Paints smooth slopes that CotNizens can walk on.
 - **Paint Cliffs** - Paints steep slopes that CotNizens cannot scale.
 - **Paint Cliff Paths** - Paints a smooth slope onto a cliff so that CotNizens can climb up the path.
5. Left-click to raise or lower the elevation of the land under the brush. Hold the mouse button and drag to “paint” that elevation level on the map. All the tiles under the brush move up (or down) to the altitude specified by the Terrain Elevation slider.

TIP: Click the “Sample Elevation” button before left-clicking to override the elevation slider. This sets the elevation level to the elevation that is currently under the center tile of the brush. That elevation setting can then be subsequently used for painting. Note that once this button is clicked, its text changes to “Sampling...” and never changes back. Once an elevation is sampled, the button becomes inoperative unless clicked again. The text will not reset until the Editor is reloaded.

Another way to create elevations on the map is to drag land up or down by clicking and holding the right mouse button and then dragging the pointer up or down. Tiles beneath the brush raise or lower as the mouse is dragged (**Note:** the choice of Paint Hills, Cliffs, or Paths is still active). If the left mouse button is simultaneously held, the elevation level being drawn at that time is then used as the current elevation until the left or both buttons is released.

NOTE: Adding hills near existing cliffs can “replace” the cliffs with smoother slopes. A little practice will help in mastering this technique.

River Creation

Once any elevation is drawn that should be under water at Normal, Low, or High Flood, the designer should create the river to ensure there are no errors. This should be done early and often in the map design process to prevent hours of wasted time.

When ready to check the river, click the **Create River** button. This will open a dialog box with another **Create River** button. Click this. Depending on the size of the map, the complexity of the river edges, and the speed of the design computer, river creation can take short seconds or a few minutes. It also can crash the editor. Saving the scenario before creating the river is advised.

One of two possibilities will occur. If **Model OK** appears, the river has been successfully created. Click **Close**, save the scenario, and continue designing, taking care to avoid editing the edges of the river unless absolutely necessary. If adjustment of river edges is necessary, save first, and create the river to check for errors, saving after each adjustment. This will absolutely save time in the long run.

If **Model Errors** appears after creating the river, then there are mistakes in the river edges. In my experience, this is most often caused by a pinch of land that does not allow a complete river flow path from map edge to map edge. Look for such an area, adjust, and try again. If the errors cannot be corrected, revert back to the earlier save and try again. This *must* be resolved or the scenario will malfunction. Start over from scratch if necessary. For this reason, I recommend that river creation be the third step of scenario creation after map generation and terrain elevation.

Once the river is successfully created, the river level can be adjusted in two ways. First are the three radio buttons that set the river to Low, Normal, and High Flood level. Selecting any of these will automatically set the river to that level. The second method is to use the **Flood Level** slider to manually adjust the river level. This is important to see which areas will be submerged by the river under various flood levels. Make minor adjustments to the river as necessary to smooth out the edges and ensure that only the areas desired to be under water are submerged.

Terrain Page

Many variations on terrain exist in CotN, and any of them may be painted by the scenario designer. Three options exist on the **Terrain Page**: **Terrain**, **Remove Sprites**, and **Paint Roads**.

49 types of terrain are available in CotN, from cliffs to scrub, and including various sprites to decorate the landscape. Additionally, 21 terrain colors exist, allowing for a vast amount of customization. Play around with these options, but note that sprites, such as weeds, exponentially increase the demand on the computer processor, slowing down the scenario considerably when used excessively.

Terrain

To paint terrain on the map:

1. Click the **Terrain** button.
2. Select the desired **Brush Size** (note that when painting terrain, the brush options are 2x2, 4x4, 6x6, 8x8, 10x10, and 12x12).
3. Choose the desired **Terrain Type** from the list.
4. Left-click on the map and drag to paint with the selected **Terrain Type**.

When painting terrain textures, remember that it is not the tiles which are being painted, but rather their vertices (that is, where the tiles intersect). This is to ensure the different terrain textures blend smoothly together. However, some Terrain Types actually “paint” units on to the map (e.g., the various Forest types). These units get placed on the tiles themselves, not their vertices.

There are six terrain types beginning with the word “Decorative.” These serve to add an appropriate base texture, e.g. “Desert,” with appropriate sprites on top of the base texture, such as rocks and desert trees. This can be a good way to add a nice area of land.

When adding sprites or using the “Decorative” options, it helps to check the “Randomness” checkbox. This will randomly rotate the sprites as they are placed, making for a less uniform placement of objects such as trees, and really breaking up the repetitiveness of the landscape.

Remove Sprites

When checked, the paint brush will remove any sprites or units on the screen under the brush. The brush settings for this option are 1x1, 3x3, 5x5, 7x7, 9x9, and 11x11.

This option is helpful if it becomes necessary to clear an area of resources quickly once they are placed on the Units Page below.

Paint Roads

When checked, the paint brush will paint the selected road type onto the map, covering all tiles under the brush. To remove painted roads, with “Paint Roads” checked, hover over the area desired to be removed, hold down the SHIFT key and left-click the tile. The road texture should be removed.

Units Page

Use the Units page to place all kinds of objects on the map, including units, buildings, and resources. Anything placed on the map in this way will be available to the player from the start of the scenario. Note that some objects cannot be placed in certain areas (for example, a CotNizen cannot be placed in the water) and objects cannot be placed on top of each other.

To place objects on the map:

1. Select **World level** to place objects that belong to the world (and not to the player), such as trees, animals, and resource sites. To place objects for the player, select **Players** to see the available objects.
2. Select the category of objects desired using the buttons at the top left of the screen. The options are: **Land** (animals and CotNizens), **Air** (this is empty in CotN), **Water** (boats and ships), **Buildings** (ambients, quarries, and player buildings), and **Resource** (all used resources in the game). The various objects belonging to the selected category will appear in the list on the left side of the screen. Note that some items must belong to the **World level** and some must belong to the **Player**.
3. Select the desired type of object from the pick list by clicking on it (e.g., click on “Archer”). The Sort buttons determine how the list of objects is sorted: **Alphabetically** (“Alpha”), by **Period** (this has no effect; all units are from Period 1 in CotN), or by **Family** (this only has an effect on the **Buildings** -> **World level** category, where quarries are grouped in Family #3).

4. To place the selected object simply left-click on the map. As many of that type of object as desired may be placed or a new object may be selected. If an object is tinted red it means it cannot be placed in that location; try a different location. Right-click to cancel the placing of that object.

TIP: To delete an object from the map, first right-click once to “dismiss” the currently chosen object. Then right-click on a placed object to delete that object.

Other Unit Controls

- **Select** - To select units already placed on the map, use any of the normal selection methods (e.g., left-click on or lasso units). If placing objects, be sure to first right-click to “dismiss” the selected object before selecting other objects on the map.
- **Delete** - Right-click on a placed unit to delete it. The Delete key on the keyboard also deletes the selected unit. To delete multiple units at once, select the units and press Shift+Delete.
- **Moving Units** - Double-click on a placed object to “pick it up.” It can then be moved to a new location (click to place it back on the map).
- **Rotate** - Left-click the Rotate button (the grey arrow) to turn a selected object clockwise or right-click to turn it counter-clockwise. The Rotate button can also be used to rotate objects about to be placed on the map. For finer control of an object’s facing, place the object(s) on the map, then select the object(s) and right-click and drag to change the direction it is facing. (Note that this only rotates the object(s) in place; it does not cause the object(s) to move.)

NOTE: do not place a “Camera” unit on a map. These are visible while playing the scenario, and camera actions can be scripted that make them unnecessary.

Triggers Page

The Triggers page allows the creation of “Triggers,” which control actions and events that take place in scenarios. There are four “components” from which Triggers are constructed. The components for Triggers are: Conditions, Effects, Objects, and Areas. These components, once created, ultimately combine together in a “Trigger Statement” to define a Trigger.

- **Trigger** - A Trigger is a set of Conditions and associated Effects. All Triggers are evaluated periodically by the CoTn game engine. If a Trigger’s Conditions are evaluated to be TRUE, then the Trigger “fires” and its Effects are queued to happen in the game. If the Conditions are evaluated as FALSE, then the Trigger does not fire at that time but remains active to be reevaluated.
- **Condition** - A Condition is a specific statement about something in the game which can be evaluated to be either TRUE or FALSE. For example, a Condition might be: “Player 1’s Pharaoh Age is greater than or equal to 20 years.” If Player 1’s Pharaoh Age equals or exceeds 20, then this Condition would be evaluated as TRUE. If Player 1’s Pharaoh Age is less than 20 the Condition is FALSE.
- **Effect** - An Effect is the result of a Trigger; it is what happens in the game when the Trigger “fires.” An Effect can be an obvious action or event that the player sees on screen, such as a displayed message or explosion or victory, or it can be something more subtle or behind-the-scenes,

such as an increase in a quantity of goods on-hand, the spawning of an enemy raid, or the activation of another Trigger.

- **Object** - An Object is a unit or list of units (or buildings) that can be combined with a state such as "has 100 Wheat" or "is near a Townhouse owned by Player 1." These units may be specific ones selected on the map or a general-purpose description of a unit, such as "any Bricklayer that belongs to Player 1."
- **Area** - An Area is a region of the map. An Area can be a rectangular region of tiles or "off" the map.

To define a Trigger or one of the four components, click on the appropriate button and that screen will appear. (For example, to define an Area, click on the Area button and the Area screen will appear.) For Triggers and each type of component, several options exist: Create a new one, Duplicate an existing one, or Delete ones no longer needed. The lists of Triggers and components can also be reordered and each can be given a unique name.

- **New** - Click the New button to create a new Trigger or component (e.g., a new Object or new Effect, etc.). This button must be clicked any time a new Trigger or component is desired.
- **Duplicate** - Select a Trigger or component in the list and then click the Duplicate button to create an exact copy of it. The duplicate is given the same name as the original with "_Copy" at the end.
- **Delete** - To delete a Trigger or component, select it in the list then click the Delete button. Note that this **permanently removes** that Trigger or component and there is no undo. Also, be careful when deleting components that are already being used in Trigger Statements as doing so may cause unexpected results in the scenario.
- **Reordering** - To the right of each component list are up and down arrows that allow moving a component up or down within its list. Simply select a component in the list and click on the arrows to move it up or down. The list can also be alphabetized by pressing the topmost button with both an up and down arrow on it. (Note that this option is not available for the Trigger list because the order of the triggers is important - see the following section on Triggers.)
- **Unique Number** - All Triggers and components are given a unique index number to aid in their identification. To enable display of this number, click the checkbox below the up and down arrows to the right of the component list.
- **Name** - The selected Trigger or component can be given any name by typing a name in the Name text box. Default names, called "autonames," are shown in triangular brackets ("< >"). It is recommended that all Triggers and components be given names that are meaningful so that they are easy to identify.
- **Hide UI** - The checkbox just above the Test button (at lower-right) shows or hides the upper portion of the UI on the Triggers page. This is useful to allow viewing more of the map at one time.
- **Description** - The Description field is a place to enter notes or comments about what each defined Trigger or component is supposed to do. It is used only for reference.

TIP: To help navigate through many components, the designer can "jump" to a specific component by right-clicking on it. For example, in a Trigger definition, right-click on a specified Condition to immediately jump to the definition of that Condition. Then right-click on a specific Object within that Condition to jump to that Object, and finally, right-click on a specific Area within that Object to jump to that Area.

Component Names and “Autonames”

Meaningful names are critical for the easy to identification of Triggers and components. For convenience, all Triggers and their components are automatically given “autonames,” but it is recommended that these be changed to names of significance. All autonames are enclosed in triangular brackets, for example “<Effect1>” or “<Area2>”, so that they are easily distinguishable from the names entered by the designer.

Trigger

Triggers are made up of Conditions and Effects, which in turn are made up of Objects and/or Areas. To create a Trigger, therefore, one must first create Conditions and Effects. (Optionally, Objects and Areas may be created, and from those components Conditions and Effects may be created.) Then combine the Conditions and Effects in a “Trigger Statement” to define a Trigger. Every Trigger must include at least one Condition and one Effect.

Triggers are evaluated periodically as the scenario is played. If and when a Trigger’s Conditions are evaluated as TRUE then the Trigger “fires” and its Effects are put into a queue to be carried out in the scenario. Otherwise the Trigger remains active and gets evaluated again later. Active Triggers are evaluated once every half a (game) second.

NOTE: A game second is not necessarily equivalent to a real-time second. The length of game seconds depends on the game speed setting selected by the player.

Trigger Order

The order of the Triggers in the trigger list is important. Triggers are evaluated from the first in the list to the last. The order in which Triggers are evaluated can be chosen by selecting a Trigger in the list and then clicking the up or down arrow buttons next to the list. The checkbox beneath the arrow buttons toggles the display of the unique number that was assigned to each Trigger when it was created.

To understand why the order of Triggers is important, consider the following example. Suppose Trigger1 creates 4 archers and Trigger2 orders the 4 archers to move to a specific area. Trigger1 must therefore appear before (above) Trigger2 in the trigger list or else Trigger2 won’t have any archers to act upon.

Trigger is On

New Triggers are automatically set to ON by default (i.e., the box is checked). Once a Trigger fires, it is automatically turned OFF (unless Looping is set ON - see below). A Trigger can be turned OFF manually (uncheck the box) to keep it from being evaluated if, for example, it is desired to have another Trigger turn it ON later in the scenario.

Conditions True For

This control sets how many game seconds the Condition(s) of a Trigger must be TRUE before the Trigger will fire. The default value is zero (0), meaning that the instant the Conditions are satisfied, the Trigger will fire. The Effect(s) of the Trigger may still be delayed; see the Trigger Statement section, below.

Looping

A Trigger can be set to “loop” so that it is always active. A looping Trigger does not turn OFF after it fires.

- **ON** - The Trigger is constantly reevaluated even if it has already fired. The Effect(s) of the Trigger are carried out every time the Trigger’s Conditions are evaluated to be TRUE.
- **OFF** - The Trigger is evaluated over and over until its Conditions are satisfied (i.e., they are TRUE), then it fires once and is never reevaluated (unless it is turned back ON by another Trigger).

TIP: Be aware that weird things can happen if looping is turned on. Since the Trigger remains active, the Effect(s) will be carried out every time the Conditions are TRUE. So, for example, if the Effect is to spawn Raiders, your map might soon be full of Raiders! Once fired, a looping Trigger is marked as “Has Fired.” Another Trigger can check to see whether a looping Trigger “Has Fired” and then turn the looping Trigger off so it is not evaluated again.

Include in Cinematic

This checkbox is a carryover from Stainless Steel Studios’ Empire Earth and has no function in CoN.

Trigger Statement

The Trigger Statement is what defines the Trigger. It is an “IF... THEN” statement of the form “IF Condition0 THEN Effect0.” To create a Trigger Statement (and thus define a Trigger), choose the Condition(s) and Effect(s) for it via drop-down lists (labeled “If” and “Then,” respectively). These drop-down lists contain all the Conditions and all the Effects that have been created for the scenario thus far, plus some defaults. “Always True” is the default Condition and “Do Nothing” is the default Effect; they are placeholders.

NOTE: Order is important. The game evaluates each Condition in the order in which it appears in the list, from left to right. Then, if the Conditions evaluate to TRUE, the game executes each Effect in order (if they have the same time delay, see below for more details on delays), from left to right.

Operators

Each Condition and each Effect in a Trigger Statement has an operator associated with it—when a Condition or an Effect is selected, an operator can be chosen for that component. Conditions and Effects can be combined using the operators AND or OR. The END operator indicates the last Condition or last Effect in the Trigger Statement. The AND and OR operators are chosen via drop down lists between the Conditions and Effects. The NOT operator is activated via a checkbox next to the Conditions list. For each Effect, a Delay to be counted before execution (in game seconds) can be specified.

The operators obey the following rules:

- **Not** - Each Condition in the Trigger Statement may be assigned a NOT operator, if desired. The NOT operator applies only to those Conditions to which it is assigned, not to

every Condition in the Statement. The NOT operator returns a value of TRUE for its associated Condition when the Condition is NOT met.

- **Delay** - A delay tells the game to wait the specified number of game seconds before carrying out the Effect to which the delay is associated. The delay timer starts when the Conditions are evaluated as TRUE and the Trigger fires. This is considered time zero (0). Therefore, if different Effects in a single Trigger each have delay times set, the delays are all relative to that zero time, and not relative to one another. (Note that, once queued up, an Effect cannot be canceled even if there is a specified delay.)
- **And** - Any Condition in a Trigger Statement can be assigned an AND operator. If the Condition preceding the AND operator is FALSE, then the Condition(s) following the AND operator will not be evaluated. Effects can also be chained together using AND.
- **Or** - Any Condition in a Trigger Statement can be assigned an OR operator. The OR operator logically joins all the Conditions preceding it with all the Conditions following it (until another OR operator is encountered). If the Conditions preceding the OR operator are TRUE then the conditions following the OR operator will not be evaluated. For Effects, the OR operator can be used to "randomize" the Effects of a Trigger (e.g., play effect1 OR effect2 OR effect3).
- **End** - End is used to signal the end of the list of Conditions and the end of the list of Effects in the Trigger Statement. It is associated automatically with the last Condition and the last Effect in the Trigger.

TIP: There are no parentheses available for grouping Conditions or Effects together in Trigger Statements. Such associative groupings can be created only by using the OR operator, so an expression such as "Condition1 AND (Condition2 OR Condition3)" can be expressed as "Condition1 AND Condition2 OR Condition1 AND Condition3". The Conditions in a Trigger are always evaluated from left to right.

When either AND or OR is chosen, a new Condition or Effect is automatically added to the Trigger Statement. As many Conditions and Effects can be contained in a Trigger as desired, but bear in mind that the longer a Trigger Statement is, the longer it will take for the game engine to evaluate it. Many such long Triggers in a scenario can cause lag in the game.

Deleting Conditions and Effects

Conditions or Effects can be removed from a Trigger Statement by deleting them. Only two Triggers and Effects can be shown on the screen at a time. To view more Triggers or Effects, use the arrows that appear above IF and THEN and above the far right AND or OR selection box to scroll to the appropriate area. Above the conditions are buttons that say Del Condition 1, Del Condition 2, Del Effect 1, or Del Effect 2 as appropriate. When the correct Condition or Effect is viewable on the screen, click the correct "Delete" button to remove the item from the scenario.

Inserting Conditions and Effects

Unfortunately, Conditions and Effects cannot be inserted into the middle of a chain of items. To add new items, they must be appended to the end of the list. To make the list more readable, it is necessary to manually resort the Conditions or Effects.

Condition

Clicking the Condition button on the Triggers Page displays the controls used for creating Conditions. Conditions are the part of a Trigger Statement that get evaluated to determine whether or not the Trigger should fire. The default Condition is "Always True," which is useful if a Trigger should fire as soon as the scenario starts. The details of a Condition—the Condition's "definition"—are entered using the controls under "Condition Details." There are several different types of Conditions, chosen from a drop-down list. Their uses are described below.

Object Exists

An "Object Exists" Condition is created automatically for each Object created for the scenario. When Objects are added, deleted, or renamed on the Objects screen, their corresponding "object exists" Condition is modified accordingly. Since there may be many of these Conditions, they are not displayed on the Conditions screen, but are available on the Trigger screen in the Conditions ("If") drop-down list.

"Object Exists" is a general-purpose Condition that tests whether the defined Object is currently in the defined state. Use it to evaluate the state of any unit, building, resource, animal, etc. defined as an Object on the Object screen. For example, one could test to see whether a specific ColNizen has entered a specified area or if a particular building has been completed. (See the section on Objects for more information.)

Conditions are grouped into types, for which the Condition Details change as appropriate. The Condition types are Player-Attribute, Trigger-Attribute, Game-Attribute, World Site, Cheat Message Contains, PlayerEventHappened, and Random.

Player-Attribute

Conditions can be defined that apply only to the player. For each such Condition, choose the Player 1 via the drop-down list under Player Attribute Condition Details.

Player Has Attribute

With this Condition, one can test whether the player has a certain amount of a given attribute. For example, one might set a condition to test whether the player has a certain amount of gold. The different options for this player Condition are explained below.

Attribute Grouping

This is the category of attribute that is to be evaluated. Beneath each are particular attributes applicable to that grouping. For example:

- **Resource** - Raw materials and finished goods, such as Wheat, Monkeys, and Turquoise Jewelry.
- **Building** - All player buildings, decorations, and monuments. Note that for each pyramid, there are two options: the pyramid base and the completed pyramid. These can be used to determine if the player has placed the pyramid construction site or if construction is completed.
- **Other Stats** - Pharaoh Age, Prestige, and 10 Player Variables (0 through 9). These attributes can be used to determine if Prestige has reached a certain level, if the Pharaoh has crossed an age limit, or to assign custom variables. (Variables can be set using Effects. See below.)

- **Variety** - Tests to determine if a variety of finished goods exists. For example, there are many types of Jewelry available. Using this attribute, one can test to see if the player has acquired 3 types in the nome.

Modifiers

Choose an inequality or equality symbol for the Condition.

- < - Less than.
- <= - Less than or equal to.
- > - Greater than.
- >= - Greater than or equal to.
- = - Equal to.

Use Amount

Use Amount compares the attribute with a specific number entered. For example, if the player's gold is greater than 500 or if the number of small pyramids is greater than 3.

Trigger-Attribute

Use this Condition to specify the state that a Trigger must be in for the condition to be TRUE. For example, if Trigger2 must fire only after Trigger1 has fired.

The Trigger

This is a drop-down list of all the existing Triggers in the scenario.

Is On

If checked, this means the Condition is TRUE if the specified Trigger is active—that is, it either hasn't fired or it's a looping Trigger that remains active after firing. (A Trigger can be created to turn other Triggers ON or OFF. See the Effects section.)

Has Fired

Once a Trigger is fired, it is always in the state of "has fired." Check this box to evaluate whether the specified Trigger has fired. (A Trigger can be created to reset a particular Trigger to "not fired." See the Effects section.)

Game-Attribute

These types of Conditions are defunct. They have no function.

World Site

These types of Conditions allow the designer to check for the status of sites on the World Level Map (See World Level Map). Choose the desired World Level Site and Condition under World Site Condition Details. The Conditions are:

Is Open

Checks to see whether the selected Site is open or activated. If the site was opened by the player by sending required materials or opened by a Trigger, this Condition will return TRUE. Otherwise, it returns FALSE.

Is Closed

Checks whether the selected Site is closed or deactivated. If the site has not been activated by the player or a Trigger closes the site, this Condition will return TRUE. Otherwise, it returns FALSE.

Is Visible

Checks whether the World Site is visible on the World Level Map. If it has been made visible by a Trigger, this Condition returns TRUE. Otherwise, it returns FALSE.

Is Dead

If the selected Site has been closed militarily by the player, or set to Dead by a Trigger, this Condition returns TRUE. Otherwise, it returns FALSE.

Cheat Message Contains

Enter a string in the text box provided. If this string appears anywhere in any chat message entered by the player during the scenario (by pressing ENTER and typing), then the Condition is TRUE. The string entered is not case sensitive, but it must match character-for-character a string somewhere in the chat message. For example, if the Condition is looking for the string "wood" and the player enters the chat message "Give me 1000 Wood" or "The woodsman" while playing the scenario, then the Condition would be TRUE. But if the Condition is looking for the string "give wood" and the player enters "Give me 1000 Wood" then it does not match exactly, so the Condition would be FALSE. This Condition is only useful for checking whether the player enters a cheat code, hence the name. It can be used to provide an effect on entering a cheat message such as instant defeat. Also, it could be used to provide Easter eggs to clever players.

PlayerEventHappened

This Condition checks whether any of a huge list of player events have taken place, such as Bilharziasis. If the selected event has occurred, the Condition returns TRUE.

Random

Allows the designer to enter a random number between 1 and 100, which the scenario will periodically evaluate. If the random number generated equals the number entered by the designer, the Condition will return TRUE.

Effect

Click the Effect button to display the controls for defining Effects. Effects are the results of Triggers; in other words, they are what happens in the scenario when a Trigger is fired. A delay can be used to keep an Effect from happening for a specified number of game seconds after the Trigger fires (the default delay is zero (0) seconds). There are several different types or "classes" of Effects, which are explained in the following sections. The default Effect for any new Trigger is "Do Nothing," which, appropriately enough, does nothing. This placeholder should be replaced with an Effect that performs some real function. Otherwise, the Effect will be mere overhead, unnecessarily slowing the scenario.

Effects come in several different types, much like Conditions. They are: Object, Player, Trigger, Media, Game, and World Site.

Object

This class of Effects deals with Objects, such as creating or removing units in the scenario. Recall that an Object is a CotNizen, building, or ambient (such as rocks) and an associated state (see the Objects section for more information about Objects). The various Object Effects (collectively called "Actions") described below can apply to any defined Object.

Create

Creates a specific unit at the location specified in the Object definition. The location can be relative either to another defined Object or to a defined Area. When an Object is created, the In Area and Has Attribute modifiers are checked to determine where to place the Object on the map (See the Objects section for more information).

Freeze

Removes AI from the specified Object. The player may still select frozen unit(s) or building(s). Frozen Objects perform no actions. Frozen units will still "fidget" (i.e., they do not literally "freeze") and can still be affected by other Triggers.

Unfreeze

Restores AI to a frozen Object. The unit(s) or building(s) return to normal function.

Kill

Kills the specified unit(s) or building(s). The appropriate death animations and sounds play when a Kill Effect is used on a building or a military unit. (When used on a non-military unit, the CotNizen will die, producing an audible sound, but no death animation will occur.) Killed Objects are removed from the scenario, but note that units Killed remain on the map; either the corpse, or in the case of non-military units their upright body.

Remove

Instantly and permanently removes the specified unit(s) or building(s) from the scenario, so all specific information about them (e.g., their current goods inventory) is lost. No animations or sounds are played. This could be used, for example, to simulate a unit entering a building or leaving the edge of the map.

Task Object

This Effect orders the unit(s) specified in an Object to move to another defined Object (i.e., another unit or a building). The default task or action, Normal, should be selected in the drop-down list (the remaining items are carryovers from Empire Earth and are defunct).

NOTE: Use of this Effect unfreezes the specified unit if it was previously frozen.

Task Area

Task Area works the same as Task Object except the unit(s) move to a specified Area. (Defining Areas is described in the Area section.) Objects can also be moved to the pre-defined "off map" Area, which is like a storage area for units in the scenario. Tasking units "off map" removes them from the map but does not delete them from the scenario. To put them back on the map where they came from, use the Previous Location pre-defined Area or any

other defined Area on the map. But note: **if there are no unoccupied tiles in the Area specified, then the unit(s) will not come back; they remain off-map.**

NOTE: Use of this Effect unfreezes the specified unit if it was previously frozen.

Unit Attribute

Allows modification of the attributes of the specific unit(s) or building(s) defined in the Object. For example, wares on hand in a house or how much wheat bread a CotNizen is carrying can be modified. Choose the desired attribute via the drop-down list. That attribute can be adjusted up or down by a set value or a percentage, or an exact amount can be specified.

Face Object

This Action Effect has no function. (Actually, it can be used, but only takes effect when CotNizens are not frozen, in which case they have AI and will go about their business, making them very difficult to control in this way.)

Player

The Player class of Effects allows the designer to change aspects of a scenario with regard to the player. These Effects, explained below, include awarding victories or defeats, scripting the camera, and allocating resources.

There are six (6) available Player Effects. They are: Victory, Defeat, Script-Camera, MiniMap-Flare, Event, and Set Variable.

NOTE: When using the Player Effects, always ensure that the Player selected is Player 1, for obvious reasons.

Victory

Immediately awards Victory to the player, ending the scenario.

Defeat

Immediately ends the scenario with a loss for the player.

Script-Camera

The Script Camera Effect can be used to control the 3D camera and create in-game cinematics to tell a story. Use the following controls to:

- **Move Camera** - Choose either Use Area or Use Object and then select any Area or Object that has already been defined from the drop down list. In the scenario, this moves the camera to the specified Object or Area (for large Areas, it moves to the center of the Area). To "turn off" Move Camera for a given Effect, which makes the camera rotate in place instead of move, choose Off Map or Previous Area for the Area or <None> for the Object (Note: any of these three options will cause the camera to rotate to face north).
- **Face Camera** - Choose either Use Area or Use Object and then select any Area or Object already defined from the drop down list. This causes the camera to rotate to face the Area or Object specified. (Again, for large Areas, the camera faces the center of the

Area.) To “turn off” Face Camera for a given Effect, which keeps the camera from rotating, choose Off Map for the Area or Empty Object for the Object (Note: any of these three options will cause the camera to rotate to face north).

- **Follow** - With the Follow box checked, the camera follows the Object specified under Move Camera.
- **Track** - If the Track box is checked, the camera continues to face the Object or Area specified under Face Camera even if the Object moves. (Note: if both Follow and Track are checked, the Camera will follow the Object under Move Camera and continue to face the Object or Area under Face Camera, allowing a panning effect.)
- **Scroll Speed** - Set the speed of the camera’s movement using this slider. Snap means the camera cuts immediately to the new location and/or angle. Or, you can choose a number from 1 to 99 to set the speed of movement with 99 being the slowest.
- **Zoom Level** - Use this slider to set the level of the zoom from -50 to 50. Or choose “Current” to use whatever the current zoom level is for the specified player(s). 50 is a wide shot of the ground from high in the air, 0 is a ground-level close-up of the specified Object or Area, and -50 looks up towards the sky from ground level.

MiniMap-Flare

This is a carryover from Empire Earth and has no function in CotN.

Event

The Event Effect can be used to trigger any situation in the game that affects the CotNizens of the nome, be it medical, dissatisfaction, et cetera. To do so, simply select the Event category, ensure that Player 1 is selected, and choose the appropriate situation. This Effect is critical, for it is the only available method of spawning raiders for the nome. Select Raiders X as the Event, where ‘X’ is the type (e.g. Greek, Asian, et cetera).

Set Variable

This Effect allows the designer to increase, decrease, or set the exact amount of a particular attribute for the player. There are ten (10) variable attributes that can be altered. This can be used innovatively to simulate nearly any condition imaginable.

Trigger

This class of Effects modifies preexisting Triggers. With it, specified Triggers can be turned ON or OFF, or their states can be set to “Fired” or “Not Fired” regardless of whether they have been already fired. Triggers can be essentially “reset” using this Effect. Also, certain Triggers can be kept inactive until they are needed in the scenario, by using another Trigger to turn them ON, thus minimizing the number of Triggers that must be evaluated by the game at any given time (this can significantly improve scenario performance).

Trigger

This is a drop-down list of existing Triggers. Choose the desired Trigger.

State

Allows selection of the state of the specified Trigger. Four states exist:

- **On** - Turns on the specified Trigger so it gets evaluated either continuously, or according to its Conditions True For checkbox.

- **Off** - Turns off the specified Trigger so it will not be evaluated.
- **Fired** - Causes the specified Trigger to be evaluated as having been fired regardless if it has fired automatically. Once a Trigger is set to the "fired" status, it is always in the state of "has fired," although it can be reset using "not fired." (Note that setting a Trigger to "fired" does not actually fire the Trigger, but only sets its status.)
- **Not Fired** - Causes the specified Trigger to be evaluated as never having been fired. This is the only way to fully reset a Trigger once it has fired.

Media

This class of Effects is used for playing sounds and music, sending messages and dialogue, and changing storyline text. Such Effects are often used for progressing the scenario's story or for giving a player instructions or hints during the scenario.

Play Sound

Use this Effect to play a particular sound, manually entered in the text box provided. Default sounds are available; if a custom sound is desired, the sound file must be in .mp3 format and saved in the data\Audio\Sounds directory under the Children of the Nile root directory. This Effect could be used, for example, to provide voiceovers (without text) or special audio cues in a scenario.

Other CotN default sound files may be used by entering correct navigation paths. For example, to play a CotN music file, first enter "..\" to move up one directory, then add "Music" to change to the Music directory, and finally, select the appropriate file. Try it with Lose.mp3. The total path is ..\Music\Lose.mp3.

Note that when playing sounds, the background music will still play if the player has it enabled. This can create some unusual effects if music is played.

TIP: When two MP3's are played back-to-back, it is possible for the second one to begin before the first has finished. To ensure that your MP3s do not overlap, it is recommended that you specify a delay time in your Trigger Statement for the Effect that plays the second MP3. But beware! Time passes more quickly in-game if the player accelerates the game speed. If the game speed is set at 2.0, a 3 second audio file will take 6 game seconds to play. Plan accordingly!

Change Text

Changes the text entered on the Overview, Tips, or Scenario Name page (see Story Page, later in this manual). Use either Add Line to add new text to the end of the existing text, or select Clear to delete any previous text. Adding a line will automatically insert a carriage return, so the next line of text will appear on its own line.

This Effect can be used so add new objectives or hints for the player over time.

Send Dialogue

Allows sending of dialogue messages to the player. Dialogue messages can appear in several different ways.

- **Message Title** - Specify a title for the message. This can be used as the subject of the message or for dialogue tags.

- **Sound (path & extension)** - As in Media, Play Sound, this will play an audio file. However, the full path must be provided here, below the data folder in the Children of the Nile root directory. For example, to reach the "misc" folder, the path must be "Audio\Sounds\misc\," without the quotes, followed by the sound file.

TIP: This entry has no function unless the option "Display" is set to Significant. (See Display, below.)

- **Message** - Type the text of the dialogue message in this field. This is what is displayed to the player.
- **Display** - This sets the type of message to be displayed. There are four types:
 - **History Only** - Displays the message only in the Message Log. The player must open the Message Log to view the information. This can be helpful for passing non-critical information to the player.
 - **Standard** - Displays the message in the status bar and the message log. The player can see a portion of the message appear on the status bar at the bottom of the screen, and can view the entire message in the Message Log.
 - **Significant** - Displays the message in a dialog box that pops up in the top-center of the screen. The text is also displayed in the Message Log.
 - **Tutorial** - Displays a special Tutorial message window where the messages are shown. This is the same window used in the CoN tutorials. The player may scroll back and forth through the messages and minimize the window.
- **Category** - These options appear to have no effect.

Send Chat Message

Sends a chat message, which appears top-center of the screen, to the player.

- **Player** - Specify Player 1 (noticing a trend here?).
- **Duration in Seconds** - This sets how long the message stays visible on the screen. Be sure to give players enough time to see and read the message. Note that game seconds pass faster than real-time seconds if the player increases the game speed.
- **Message** - Type the chat message to send in this field.

TIP: To make it appear as though a chat message is coming from a particular unit, manually type in a suitable name in the Message text field (e.g., "The King: " or "Soldier 1: ") and then type the message after the name.

Game

The following class of Effects produces effects related to the game world.

Graphic Effect

Allows the playing of special graphical effects, such as an explosion or smoke or even a calamity. Select the desired special graphical effect Graphic Effect list. Next, pick the defined Object or Area where the graphic effect should appear.

TIP: play around with the graphic effects. There are many. Some do not function, other can produce spectacular effects. But beware; some will crash the scenario. Test thoroughly!

Set Next Flood

This allows the designer to set the next flood to be of a certain type: No Flood, Standard Flood, Rich Flood, or High Flood. Normal messages and conditions that would normally occur with such conditions will also occur after using this Effect. Note that the Effect has to be queued some time before the start of flood (the beginning of Shemu produces good results), or the flood will not be affected until the following year.

World Site

World Site Effects allow manipulation of sites on the World Map. These are some of the most powerful Effects in CotN, as they allow considerable latitude in storytelling and game play experience. Several options exist: Open Site, Close Site, Make site visible, Make site invisible, Kill Site, Move Site, and Spawn Caravan.

Open Site

Places the selected World Level site into an Open, or Active state. This will cause the site to begin interacting with the nome as though the player activated it with resources, and will change its appearance on the World Map.

Close Site

Places the selected World Level site into a Closed, or Inactive state. This will cause the site to cease interacting with the nome, as though it had never been opened, and will update the site's appearance on the World Map.

Make site visible

Causes the site to appear on the World Map. The site will appear in whatever condition it currently is (Open, Closed, Killed, et cetera).

Make site invisible

The inverse of Make site visible. The site will disappear from the World Map, but will remain in whatever state it is currently set.

Kill Site

This will convert the site into a defeated state, as though the player sent a successful military invasion.

Move Site

Allows the change of location of the selected site on the World Map. To use, set X Pos and Y Pos to a positive figure. The grid lays out figures for X from 0 in the west to approximately 780 in the east, and figures for Y from 0 in the north to approximately 550 in the south. The number entered in X Pos and Y Pos will move the site to the new location provided (it is not a movement by an amount of distance!).

Spawn Caravan

Allows the prearranged creation of a caravan from the selected site, carrying a specified amount of a specified good. Choose the site, then the good and amount for the caravan to bring.

Object

Clicking the Object button on the Triggers page displays the controls used for defining Objects, which may then be used in creating Conditions and Effects. "Objects" refers both to objects (e.g., units, buildings, etc.) and the possible states they might be in (e.g., in a specific area, et cetera). Objects are therefore very versatile components. They are used as qualifiers for Conditions, as recipients of Effects, and to define where an Effect takes place.

Object Autonyms

Like other Trigger components, defined Objects are given "autonyms." (It is recommended that these be changed to names that are significant.) The autonyming of Objects uses a few rules and symbols to help distinguish between similar Objects. If an Object contains a unit selected on the map, it is given an autonym of the unit's class type (Baker, Noble, Farmer, etc.), followed by the unit's unique identification number. Thus, one Object may be named "<Baker231>" and another "<Farmer112>". If an Object contains multiple units selected on the map, it is given an autonym of the first unit selected followed by a "+" symbol to indicate that there are additional units in the Object. For example, if an Object contains both a Baker and a Basket Maker, its autonym might be "<Baker231+>".

For Objects that use an Object Specification (see below), the autonym consists of the name of the class of unit used by the object. This may be additionally followed by an asterisk "*" if the Object is dynamic. If the Object Specification uses a nested Object, its autonym starts with "#:" followed by the name of the nested Object (e.g., "<#:<BasketMaker244>*").

Select On Map

This is one method of selecting units and buildings to use in an Object definition. Specific units and buildings can be chosen from the map by either clicking on or "lassoing" (clicking and dragging) them. The names of the units currently in the Object definition appear in the Unit Selected List in the upper right of the Object screen. Left-clicking or lassoing units continually adds units to the Object definition. Right-clicking or right-lassoing selected units removes them from the Object definition. Right-click on an empty part of the map to remove all units from the Object. Clicking on a unit in the Unit Selected List centers the camera view on that unit.

Object Specification

Use Object Specification to specify units (or buildings) not yet on the map to be part of an Object definition. During the scenario, the game engine searches for those units that fulfill the requirements provided and, once they are found, uses them in the Object definition. Object Specifications are defined using the following criteria (if specifying an Object by one of the following criteria is not desired, leave the default setting in place):

Owner

Again, this should always be left at Player 1.

Unit Set

This is a drop-down list of loose categories into which the unit(s) or building(s) must fall in order to be selected by the Object Specification. Families include dedicated buildings, farmers, adults, et cetera.

Class

This is a drop-down list of specific unit and building types, such as a baker, shipyard, or yacht.

Use Object

Objects can be created that refer to other predefined Objects. This is called “nesting” an Object. When an Object is “nested” inside a “higher-level” Object, the qualifiers of the higher-level Object overrule those of the nested Object, but only if the nested Object is either of the type “Select On Map” or if it is an Object Specification that is not Dynamic. If the nested Object is Dynamic, then the nested Object is evaluated with its own qualifiers in order to return a list of units to the higher-level Object. Once the list of units is given to the higher-level Object, then the higher-level Object is evaluated with its own qualifiers. (This is some heavy language. Play around with this to better understand.)

In the case where a nested Object is defined as not Dynamic but the higher-level Object of which it is a part is evaluated *first*, then the nested Object is evaluated as if it were Dynamic (that is, it returns a list of units to the higher-level Object).

Example: Object A is defined as a Cosmetician, selected on the map, with the qualifiers In Area and Selected By activated. Object B uses Object A in its definition and adds the qualifier Has Attribute with the value Emerald Jewelry min 20 and max 20. Object A is, therefore, nested inside Object B, making B the “higher-level” Object. As a result, when Object B is evaluated, only the Has Attribute qualifier is used, so the Trigger looks for the Cosmetician on the map and checks to see if it has 20 or more Emerald Jewelry. The other two qualifiers (i.e., In Area and Selected By) are ignored.

NOTE: Anytime an Object is “nested” inside another Object with Use Object, the game checks to ensure a closed loop does not exist (e.g., A is nested in B which is nested in C which is nested in A). If this is the case, nesting will not be allowed.

Also, Use Object is always superseded by both Family and Class. If either is selected, the Use Object selection is ignored.

Dynamic

If the Dynamic box is unchecked, the first unit(s) or building(s) found to meet the indicated requirements of the Object Specification will be chosen for the Object and no further selecting will be done, even if the Object is used in another Condition or Effect. Checking the Dynamic box means new units or buildings will be picked for the Object each time the Object is used (note that the same unit(s) may or may not be selected each time). This is useful if the same Object Specification is intended to be used in several Condition and/or Effect definitions but the very same unit(s) is not required each time.

Allow Incomplete

If checked, the first building under construction or complete found that meets the requirements of the Object Specification will be chosen. This can be used, for example, to select an under construction Overseer’s Quarters, to determine if it has been placed in a specific Area (see Area, below).

Min and Max

The minimum and maximum number of units or buildings to be used in the Object Specification can be set. When creating or selecting the Object during the scenario, the game engine will use at least the minimum number, but will try to find up to the maximum number of units or buildings that fulfill the requirements of the Object Specification. To cause the Object Specification to use a specific quantity, make Min and Max the same number. For example, to always choose ten farmers for the Object, set both Min and Max to 10.

Object States

The Object States are the portion of the Object definition that deal with the various states an Object might be in. All of the different Object States described below can apply to both Conditions and Effects, specifically the Object Exists Condition and the Object class of Effects. For the Object Exists Condition these states act as qualifiers in determining whether the Condition is TRUE or FALSE (for this purpose, all the qualifiers must be true in order for the Condition to be evaluated as TRUE). For the Object class of Effects they act as modifiers to describe how or where or to what the Effect should be applied (e.g., kill a Noble who is carrying no wheat bread). Use as many or as few of these states in an Object definition as desired; they are optional.

NOTE: The checkboxes for each state have three settings: OFF, ON, and NOT. OFF is represented by an unchecked box, ON is checked, and NOT is represented by an "X" in the box. The NOT setting is useful for excluding states, for example, NOT Near a specified Object or NOT in an Area.

In Area

Check the In Area box to choose an Area from among those already defined (see the Areas section in this chapter). This is the "target area" for the Conditions or Effects that use this Object in their definitions. Use In Area if seeking to confine a Condition or Effect to a specific Area of the map.

In addition to defined Areas, there are two special, predefined Areas available: Off Map and Previous Location. Off Map is like a storage area for units that are not currently on the map but still exist in the scenario. It is useful for removing units from the map that are to be returned to the map later on. The Previous Location area is used to return Off Map units to the same place on the map from where they were removed.

NOTE: If the game tries to create an Object using the In Area qualifier and the target Area does not contain any valid placement tiles, then the Object will not be created.

Distance To

Similar to In Area except a previously defined Object is selected rather than an Area to be the "target." For example, a Condition might be evaluated as TRUE once a unit gets within some number of tiles of a specified building, or an Effect can be made to unfreeze units that are within the specified distance from a specific Citizen or building. To use this option, specify the number of tiles and the desired target Object.

Has Attribute

This Object State allows specification of a particular attribute of an Object (i.e., unit or building) that “narrows down” how a Condition is evaluated or how an Effect is applied. There are many attributes available in the drop down list, such as Resources or Finished Goods being carried, as well as a few that are not used in CotN. An exact amount, a range of values, or a percentage can be set for the selected attribute. To set a range, enter a Min and a Max value. To set an exact amount, set the Min and Max to the same number. To use a percentage instead of an amount, check the % box. The percent is calculated with respect to the base amount of the specified attribute for the unit(s) or building(s) in the Object.

Bear in mind that if an Object contains units of various types, then checking Has Attribute can lead to the Object State never being evaluated as TRUE if at least some of the units in the Object do not have that attribute. For example, suppose a Brickmaker and a Farmer on the map are selected to be an Object, and the Has Attribute state is set to “Clay between 5 and 10.” Since Farmers do not gather Clay, this state can never be TRUE if it is used as an “Object Exists” Condition in a Trigger Statement. Likewise, if used in a “Kill Object” Effect, the Effect would not be carried out because no Objects in the scenario exactly fit the given description.

NOTE: Nested Objects can render the above example untrue. If Object A is defined as a Brickmaker and a Farmer, and Object B is Ref:A, Clay Min 5 Max 10, then Object B could be TRUE if the Brickmaker gathers some Clay.

Selected By

Defines the unit(s) or building(s) that the player must select in order for a Condition that uses the Object to be evaluated as TRUE. If the Object contains a group of units and/or buildings, only one member of the group must be selected by the player for the condition to be TRUE. For an Effect, Selected By could be used, for example, to remove a unit from the map if it is currently selected by the player.

NOTE: Selected By is ignored for Create Object Effects.

Area

Clicking the Area button displays the controls used for defining Areas, which may then be used in Object and Effect definitions. Areas can be defined only one way: Select Rectangular Area.

Select Rectangular Area

Allows definition of a rectangular Area of tiles by clicking and dragging the mouse pointer on the map. The selected tiles are highlighted. While selecting an Area, drag the mouse cursor off the map to select edge tiles.

Center Camera On Area

The camera can be centered on the Area currently selected in the list by clicking the Center Camera on Area button. This is a useful tool for to help remember where a particular Area is located. The camera centers on the center tile of the Area.

Description

Use this box to add a meaningful description of the Area if desired.

Players Page

On the Players page, the starting conditions can be set for the player in the scenario. This includes selecting the starting season of the scenario, allocating starting resources, choosing available buildings and deities, and other settings. The Players page is divided into several sub-pages. Each sub-page is accessible by clicking on its button at the top of the screen.

Events

With the controls on the Events tab, player events can be enabled or disabled as desired.

- **Enabled Events** - A list of all Events which can occur randomly within the scenario. These may also be executed by a Trigger Effect.
- **Disabled Events** - List of all Events which cannot occur within the scenario randomly. They may still be generated by a Trigger Effect.

Worship

All aspects of the nome's Worship can be set from this page.

- **City Patron Deity** - Select one of the gods to be the city's patron deity. Once this is set, the people will have dissatisfaction when they cannot worship in a Temple. If left at None, no concern will be given to a patron deity.
- **Enabled Deities** - The list of deities for which periodic worship needs will occur (e.g., Thoth for Scribes).
- **Disabled Deities** - The list of deities for whom no periodic worship needs will occur. This can be useful to dedicate the city to one particular deity.

Construction

Allows modification of which buildings may be constructed in a scenario.

- **Units** - Has no function.
- **Buildings** - Allows or prevents the construction of specific buildings in the scenario. Note that to prevent the construction of a pyramid, the pyramid base must be disabled. Additionally, there are several items (most notably prestige buildings) that have multiple items in the list. Only one item is required to prevent their construction, but other than the pyramid, it is not possible to determine which, so it is best to disable all of them. Finally, to enable the Great Lighthouse and Great Library (Alexandria only), they must be added to the list of enabled buildings and the "Allow Great Works Buildings" checkbox must be checked.
- **Prestige Buildings** - This section has no function.

Options

The Options sub-page gives great control over starting conditions of a scenario.

Starting Bricks - This is a very important option, as it determines what is possible from the start of the scenario. Without bricks, the first Bakery cannot be built, and dissatisfaction will be high among government workers, possibly for several years.

Starting Food - This is also of critical importance, as it affects the mood of all citizens, including the Palace Citizens, from the start of the scenario.

Year - Select a number for the year, e.g., 35 or 423.

A.D. / B.C. Selector - Use this setting to specify whether the year is B.C.E. or C.E. (A.D. or B.C.).

Week - Specify the first week of the scenario. Week 48 is the start of Shemu (Harvest), Week 14 is the start of Achet (Flood), and Week 30 is the start of Peret (Planting).

Time Period - Interesting, but has no effect on the scenario.

Current Pharaoh - Also has no effect on the scenario.

Initial Prestige - This can have a big effect. Prestige determines the number of educated workers possible, and therefore a number above 0 here gives the player an advantage.

Starting Pharaoh Age - Another potential for advantage or disadvantage. Younger Pharaohs tend not to have accidents (though not always), and older ones tend to die soon (though again, not always).

Starting Pharaoh Number - Purely cosmetic. Adds the I, II, III, et cetera after the Pharaoh name.

Story Page

This page allows entering of story information, instructions, and tips to appear on the Scenario Intro (loading) screen. A JPEG background, movie, and audio file can also be used, and one can Import or Export a localization file.

- **Scenario Inst. Map** - Choose the JPEG file for the map or graphic to display as the background on the Scenario Intro screen, where the instructions are shown. These JPEG files must be saved in the ..\Data\Scenarios folder.
- **Movie** - a custom movie can be added here. Movies must be in .bik format and must reside in the ..\Data\Movies folder. However, movies will be displayed in their default resolution, so this can produce a rather ugly effect.
- **Inst. Sound Over** - a custom sound introduction can be used, like the ones in the default CoTN campaign. Files must be in .mp3 format and must reside in the ..\Data\Audio\Sounds directory.
- **Scenario Name** - The name of the scenario, which is displayed at the top of the introduction and completion screens.
- **Scenario Instructions** - Enter the instructions for the scenario. These may include directions, objectives, story elements or whatever you want. Instructions appear at the beginning of the scenario on the Overview tab of the Scenario Information screen (before the scenario is loaded).
- **Tips** - Hints and Tips for the scenario. These appear on the Tips tab of the Scenario Information screen.
- **History** - This is not used.
- **Localization Text File Name** - Type in a file name for the localization (.lcl) file. Localization files are an output of all the text (i.e., dialogues, unit names, etc.) in the scenario.
- **Export** - Click on Export to export all text (i.e., History, Tips, etc.) to the .lcl file specified in the Localization Text File Name, located in the ..\Data\Scenarios folder. Once the .lcl file has been created, the text for a scenario can be edited outside of CoTN, for example in a text editor.
- **Import** - Import the text from the specified localization (.lcl) file into the scenario.

Debrief Page

This page allows entry of text, sounds, and movies for use at the conclusion of a scenario.

- **Movie** - a custom movie can be added here. Movies must be in .bik format and must reside in the ..\Data\Movies folder. However, movies will be displayed in their default resolution, so this can produce a rather ugly effect.
- **Inst. Sound Over** - a custom sound introduction can be used, like the ones in the default CotN campaign. Files must be in .mp3 format and must reside in the ..\Data\Audio\Sounds directory.
- **Aftermath** - the text that appears at the successful conclusion of the scenario as resolution of the story.

NOTE: The text entered in the Aftermath box appears only after a win in a scenario. The text that appears on a failure is auto-generated and hard-coded, and therefore cannot be changed on a per-scenario basis.

Goals Page

The Goals Page allows the scenario designer to create fairly extensive goals for the scenario. In addition to providing goals required for victory, one can also create goals that will guarantee defeat. Be careful with these. Think of these essentially as Conditions for Victory or Defeat. If a series of Victory (AND) Goals are entered (see below), all must be true to provide Victory, so make sure they are all possible!

Goal List

This box contains all Goals that exist for the scenario, by name only. Click on the name to see the Goal details below.

Name

Enter a name for the Goal. Make this a name of significance, as it will be the only way to identify the Goal in the Goal List.

Note that each item below is specific to each Goal entered in the Goal List, including the Goal Overview and Scenario Tips information. All text entered in those fields will be listed separately for each Goal in the Goal List. This gives a designer considerable control and structure over the scenario requirements.

Effect of Goal

Choose from the available options for the type of goal, or more specifically what the goal's result will be.

Victory (OR)

A Goal selected with an effect of Victory (OR) is an optional Goal for Victory. This means that either this OR some other Victory (OR) Goal can occur, and either will produce a Victory for the player.

Defeat

A Goal marked with a Defeat effect will produce Defeat for the player. Unfortunately there is no AND effect for defeat Goals. If multiple Defeat Goals are entered, any one of them will produce

Defeat for the player. To produce a Defeat AND series of Goals, use a Player Event Trigger with a series of Conditions.

Victory (AND)

A Goal with a Victory (AND) effect is required for victory. As long as there is at least one other Goal with a Victory (AND) effect, each and every one of those Goals must be met to provide a Victory for the player.

Goal Category

The Goal Category is a selector to establish the necessary item to meet the goal. It also determines the list in the Player Attribute for Goal field (see below). The options are Resources, Varieties, Citizens, Buildings, Attributes, and World Site, and each of them operates precisely as they do in other areas of the Scenario Builder.

Player Attribute for Goal

Here is where the designer selects the requirement to meet the Goal. Its contents are depended on the Goal Category selector (see above). Whatever is selected in this field (such as Farmhouses, or Jewelry, or Prestige) provides the metric against which the Goal is measured. If the specified quantity of this item does or does not exist, the Goal is either met or not.

Qualifier

Specifies the operator for the Goal. Options are <, <=, =, >=, and >.

Amount

The specified quantity necessary to meet the Goal.

Delay

This appears to have no effect.

Duration in Seconds

This appears to have no effect.

Goal Overview

Text entered here is appended to the end of the Scenario Instructions text from the Story Page. No carriage returns are added, so it is important to add them to make the text read as intended.

Scenario Tips

Text entered here is appended to the end of the Tips text from the Story Page. Again, no carriage returns are automatically added, so they must be inserted manually by the designer.

Reset Some Entry Fields

This would be more aptly named "Reset All Entry Fields" because that is what it does. It creates a blank Goal so that the designer can create a new one.

Delete Goal

Deletes the Goal currently selected in the Goal List.

Replace Goal

Replaces the Goal currently selected in the Goal List with a Goal matching the information as currently entered. This can be used to edit an existing Goal to correct errors, or to produce a second, nearly-identical Goal to a previously entered one.

Submit as Goal

Creates a new Goal with the information as currently entered. This is the method necessary for entering a new Goal.

World Level Page

The World Level Page is where the scenario truly comes to life. It allows all the interaction between the nome and the rest of the Egyptian world, and the highest elements of storytelling on the part of the designer.

That said, while there are many controls to deal with on the World Level, it really is quite simple once one becomes familiar with it. The creation of a World Map essentially consists of three steps: selection of a World Map, creation of a New Scenario (the home city or nome), and creation of all the World Level sites necessary for the scenario. Each of these steps is discussed below in detail, and they should be conducted in the order specified.

World Map Selection

This step determines how the World Map background will appear in the scenario. Select the map to appear under the World Map selection box. Several defaults appear in this list, but custom maps may also be generated.

To create a custom World Map, use a picture editor (Photoshop is my personal choice) to create a 1024 x 768 map, made on a canvas of 1024 x 1024. The map picture itself should be centered and top-justified, so that there is a black rectangle of size 1024 x 256 at the bottom. Once this is done, the file must be converted to .dds format (look for the .dds plug-in for Adobe Photoshop, for example), with the following options:

- Alpha, Red, Green, Blue, 8 bits per channel
- DXT3 Compression format
- Single MIP level

Then the file must be saved in .dds format, with a name beginning with "wl_map_" and followed by a name of the designer's choice. If the map is not named "wl_map_" it will cause an editor crash. The file should be saved into the \Data\Graphics\World Level\Map folder. Then after a restart of the Scenario Editor, it will be available for selection.

If such a map is used, it must be distributed with the scenario, or CoTN will use the default wl_map_1_1.

New Scenario Creation

The next step in World Map creation is the creation of the "New Scenario." This step is critical, and would more aptly be named "Home City Creation," as it creates the home spot for the scenario (the player's nome), which is used to measure distances to all spots on the map (and therefore travel times).

Click "New Scenario." The screen will change, creating several new options. Starting from the top left:

'x' Description

In this box, the designer can enter text for the three cases, Active, Inactive, and Carryover. The 'x' in this section's title is replaced with those words, depending on which button is clicked, such that the title can be "Active Description," "Inactive Description," or "Carryover Description."

The text entered here is shown in the site description on the World Map. In the scenario, the only text that will ever show is the text on the Active box. In a Campaign, if the site is carried to the next scenario, the Carryover text is used (see Campaign Editor, later in this manual).

Active

This text is displayed during the scenario for the home city, below the site name in the top left corner of the World Map. This is usually where something like "This is our city!" is shown.

Inactive

This text is only displayed for a scenario site on the flyover text when mouse hovering over the site. That said, it only appears if the site is marked as Inactive in the editor. It is possible, however, to have two scenario sites on the map, with one inactive (or both active for that matter). However, placing two or more scenario sites is merely cosmetic, as the second one cannot interact with the home city in any way, and cannot be activated via triggers. The only possible interaction would be through scripted dialogue or some other means, as to the Scenario Editor, the site doesn't exist as a World Level site.

Carryover

Text entered here appears as flyover text when mouse hovering over the site in a subsequent scenario of a campaign.

For example, the Alexandria campaign begins with Naukratis, and then proceeds to Pharos. While playing the Pharos scenario, Naukratis is visible on the World Map as a Carryover icon (the golden version of the city), and when the mouse is hovering over the site, this text is visible.

Map Label

This is the name by which the site will be recognized on the World Map. It shows on the city label on the map if labels are enabled, and in the top-left of the World Map screen. This is an essential label.

X Pos. Y Pos

The X position and Y position determine the location of the site on the World Map. The grid lays out figures for X from 0 in the west to approximately 780 in the east, and figures for Y from 0 in the north to approximately 550 in the south.

NOTE: X coordinates below approximately 225 are not visible on the World Map due to the in-game interface.

Text Align Left

This checkbox will cause the normally on-the-right text to be placed to the left of the World Map icon. This is helpful when placing sites in the East, where the label will be written off the edge of the map.

Move

This checkbox allows easy placement of the site. Check this box, then click the location on the map where the site should be located. Then click this box again to uncheck it, and then click Replace to save the change to the site (see Replace below).

Add / Replace

These buttons either add a site for the first time, or save changes made to an existing site, as their name implies.

Cancel

Cancels any changes made to a site and returns the designer to the main World Map view.

Graphic

Allows selection of the icon which is displayed on the World Map to represent the site.

Active

If checked, the site will be Active when the scenario begins. That is, it will be able to interact with the nome. For the Scenario (home city) site, this should always be checked.

Our City

If checked, the site is the site used as the home city from which the distances to all other sites are measured. For the scenario site, this should always be checked.

Carryover

If checked, the site will be shown with its carryover icon in subsequent tiers of a Campaign (see Campaign Editor below). The Carryover text is used in the later scenarios.

Scenario File

This allows selection of the scenario for which the New Scenario is applicable. Always select the currently edited scenario in this box.

New Activation Site

When the New Activation Site button is clicked, the screen changes to allow the creation of a new World Map site. These sites are those which interact with the nome and drive the story forward. They consist of trading partners, resource locations, quarries, enemies, et cetera. These sites will make up the bulk of any World Map.

'x' Description

This box works just as it does with the New Scenario button (see New Scenario Creation above). However, there are additional cases. Active, Inactive, and Carryover are still present but are joined by Deactive, Prop Build and Prop Descrip. The 'x' in this section's title is replaced with

those words, depending on which button is clicked, such that the title can be "Active Description," "Inactive Description," "Deactive Description," "Carryover Description," "Propaganda Build Button Description," or "Propaganda Description."

The text entered here is shown in the site description on the World Map for all options but Carryover. The Carryover text is shown only if the site is carried over in a Campaign (see below).

Active

This text is displayed during the scenario the site (when active), below the site name in the top left corner of the World Map.

Inactive

This text is displayed below the site name and on the flyover text when the site is set in an Inactive state. This can be done at the beginning of the scenario via the Inactive checkbox (see below), or by a Trigger at any time during the scenario.

Deactive

Text entered here appears below the site name and as flyover text when an enemy site is in a Deactive state.

Such a state can be achieved either via satisfying the conditions necessary to defeat the site or by using a "Kill Site" Trigger.

Carryover

Text entered here appears as flyover text when mouse hovering over the site in a subsequent scenario of a campaign.

This works just as the Carryover text works on the New Scenario site.

Prop Build

Text entered in this box appears on the build menu for Stele and Obelisks, but only if the site's Propaganda Value is set greater than 0 (see below).

Prop Descrip

Text entered in this box appears in the Selected Item Area when the propaganda item (Small Stele, Stele, Obelisks, and Grand Obelisks) is selected.

Resources Area

The Resources Area lays out all Imports and Exports for a site, the Activation costs, and the Deactivation (Kill) costs. Additionally, it can be used to set prestige, et cetera.

Map Label

This is the name by which the site will be recognized on the World Map. It shows on the city label on the map if labels are enabled, and in the top-left of the World Map screen. This is an essential label.

X Pos. Y Pos

The X position and Y position determine the location of the site on the World Map. The grid lays out figures for X from 0 in the west to approximately 780 in the east, and figures for Y from 0 in the north to approximately 550 in the south.

NOTE: X coordinates less than approximately 225 are not visible on the World Map due to the in-game World Map interface used to provide information about sites and goods. Be careful about site placements here.

Text Align Left

This checkbox will cause the normally on-the-right text to be placed to the left of the World Map icon. This is helpful when placing sites in the East, where the label will be written off the edge of the map.

Move

This checkbox allows easy placement of the site. Check this box, then click the location on the map where the site should be located. Then click this box again to uncheck it, and then click Replace to save the change to the site (see Replace below).

Add / Replace

These buttons either add a site for the first time, or save changes made to an existing site, as their name implies.

Cancel

Cancels any changes made to a site and returns the designer to the main World Map view.

Graphic

Allows selection of the icon which is displayed on the World Map to represent the site.

Active

If checked, the site will be Active when the scenario begins. That is, it will be able to interact with the nome. For the Scenario (home city) site, this should always be checked.

Our City

If checked, the site is the site used as the home city from which the distances to all other sites are measured. For the scenario site, this should always be checked.

Carryover

If checked, the site will be shown with its carryover icon in subsequent tiers of a Campaign (see Campaign Editor below). The Carryover text is used in the later scenarios.

CAMPAIGN EDITOR

The Campaign Editor allows designers to connect individual scenarios together to create a whole campaign, like collecting chapters together into a complete book. Campaigns can be used to tell epic tales (like the Grand Campaign found in original CotN, or the Alexandria campaign found in the expansion), with the scenarios providing the different scenes where the characters interact and the action takes place. Or, a campaign might be a collection of “short stories,” with each scenario standing as a story in its own right. The contents of a campaign are ultimately up to the designer. However, the scenarios must be created first, before they can be assembled into a complete campaign.

To play a created campaign, the campaign file must be saved in the ..\data\campaign folder under the root folder. Then start CotN, click Play, then Play a Campaign, then Other Campaigns. Choose the desired custom campaign from the list and click OK to begin.

Starting the Campaign Editor

To start the Campaign Editor, click the Editor button on CotN’s Main Screen and then click the Create Campaigns button.

Using the Campaign Editor

Creating a campaign is a relatively simple task, but one simple rule must be remembered:

1. Create all the scenarios for the campaign first and make sure they are saved in the scenarios directory in the data folder. (...\data\scenarios). The root path will vary depending on where CotN is installed on each PC.)

Once the scenarios have been saved in the correct folder, add them into the campaign in the desired order, give the campaign a name, and save it. Then this file can be distributed to players as a stand-alone package which includes all scenarios and media files.

Detailed information on using the Campaign Editor and some additional options are described in the following sections.

NOTE: Leave the Campaign Editor by clicking the Exit button.

Background Image

Select the appropriate World Map background picture. It is helpful to choose the same map as is selected in the scenarios you’ve created, for continuity. However, it is also conceivable to display an overview map at the Campaign level, and zoomed-in detail maps at the Scenario level.

Campaign Name

Enter the name of the campaign in this field. If a pre-existing campaign is loaded, its name appears here. Editing a loaded campaign’s name will force the Campaign Editor to create a new campaign based on the settings of the loaded campaign. Essentially, the act of renaming a campaign duplicates it.

To be clear, the name used here is used as the file name, as in Campaign Name.ssa. If this field is changed, the file must be saved and will use this as the file name. If, when saving the file, the file name is changed, this box will also be changed. The text in this field cannot be different from the file name.

NOTE: It is wise to limit Campaign names to 16 characters or less to prevent wrapping the text on the site selection screen.

Campaign Description (Optional)

The Campaign Description field allows the designer to enter an (optional) description for the campaign. The description can be used to tell a back story, present an overview of the campaign and its objectives, give instructions, or provide any other relevant information. Players see the description when they select Open on the Other Campaigns screen.

Campaign Tier

This box determines which Campaign tier the description text below it is used to describe. For example, if this box is set to 0 (the first level), the text entered below will be shown when a Campaign tier 0 city is selected in the site selection screen of the Campaign.

To use this box, select the tier number, then update the text. Finally, press the Update button, described below, to save the change to the Campaign. Finally, the file must be saved.

Campaign Tier Description

This box is immediately below the Campaign Tier box and provides a place for description text used at each tier of the campaign when cities are selected. Enter the text here in accordance with the instructions given in the Campaign Tier section, above.

NOTE: Text entered here is lost once the campaign is saved for all tiers above 0. When loading the campaign, the text for tier 0 is displayed, and there is no way to retrieve the text from the other tiers. If this text is important, keep it stored in a safe place.

Scenarios in Campaign

All the scenarios in the currently selected campaign are displayed in this list. They are played in order according to the Scenario Tier box, described below.

Add Button

The Add button moves the currently selected scenario in the Available Scenarios list to the Scenarios in Campaign list. In other words, clicking the Add button adds the selected scenario to the campaign.

NOTE: If an attempt is made to add a scenario to the campaign with a necessary media file (e.g., JPEG, MP3) not in the correct folder, a notification will appear that the file is missing. The campaign cannot be saved until all the media files for all the scenarios are in place. See the Scenario Editor section for information about making scenarios.

Remove Button

The Remove button removes a selected scenario from the Scenarios in Campaign list and puts it back in the Available Scenarios list.

Update Button

The Update button allows the designer to make changes to the selected scenario from the Scenarios in Campaign list. Make the necessary changes and click Update to commit them, then save the Campaign.

Scenario Information

All the items below can be changed either before clicking either the Add or Update button. Whatever is entered in each of these items will be ascribed to the scenario location at that time, but may be edited using the Update button.

City Name

The text entered here will appear on the site selection screen of the campaign. It is helpful to use the proper name of the city here, e.g. Nekhen. However, this can differ from the city name used in the scenario, if desired.

Scenario Information

Text entered here is appended to the text given in the Campaign Tier Description box, explained above. It is helpful to add a line break, either at the beginning of this text or the end of the Campaign Tier Description text, for clarity.

Scenario Tier

This box determines at which Campaign Tier (beginning at 0) the selected scenario will be available for play.

Clear

This button will clear everything from the City Name, Scenario Information, Scenario Tier, XPos, and YPos areas so that new information may be inserted.

XPos

Determines the location, from left to right, on the map where the scenario site is displayed. Appropriate values are 0 to approximately 700.

NOTE: Locations below approximately 275 will be beneath the user interface in the site selection screen, but they will appear through the interface and can be selected.

YPos

Determines the location, from top to bottom, on the map where the scenario site is displayed. Appropriate values are 0 to approximately 950.

Icon

Allows selection of the display icon, identical to the selections in the Scenario Editor for World Level sites.

Available Scenarios

This list displays all the scenarios available to add to the campaign. Scenarios must be saved in the ..\data\scenarios folder in order to be displayed here.

New Button

Click this button to reset the Campaign Editor screen to allow creation of a new campaign. Default names and text are added to some of the text fields.

Load Button

Opens a dialog box to allow loading of all available campaign files. Campaigns must be in the ...\data\campaigns folder to be displayed here.

Save Button

Click Save to save the current campaign using the name in the Campaign Name text box. The file is saved into the ..\data\campaigns directory. Saving a campaign also automatically compiles all the necessary components of the campaign, such as scenarios and media files, so that the campaign file is all a user needs to play the campaign (the file must be placed in the ..\data\campaigns folder by the user to be available).

Exit Button

Exits the Campaign Editor.

APPENDICES

Appendix 1: Trigger Example

This Appendix provides an example of creating Triggers for a scenario. This example is fairly simple. If new to the Scenario Editor, this example should help you better understand how Triggers work. As with anything, don't forget to save regularly over time!

Example

This is an example of creating a Trigger to move units using Cinematic Marker Objects. The first thing to do, of course, is to start CoTn and then launch the Scenario Editor. A new map can be generated if desired, but it is not necessary for this example; a blank map is fine.

1. First place some units on the map. Click the Units button at the bottom of the screen. The default settings on the Units screen are: Land and Players, and the list of units is sorted alphabetically. From the list on the left, select Chariot Warrior and then left-click on the map to place a unit at that location. Place four Chariots in total in a straight line.
2. Change to the Triggers page and add four areas, a bit out in front of each of the four Chariot Warriors. Rename these to Area 1, Area 2, Area 3, and Area 4.
3. Now it's time to start creating a Trigger. The first step is to create Objects. Click the Object button (labeled "Object"). Click "New" to create a new Object. Be sure "Select On Map" is selected and then click on one of the Chariot Warriors. This Warrior has now been specified as an Object on the map. This specific Object can be used in numerous ways, some of which will be explained below. Create a new Object for each Warrior in the same way—by first clicking "New" then selecting the next Warrior on the map.
4. It is recommended that each Object be renamed so it's easy to remember what each one is. Select the first Warrior in the list of Objects (it will be called something like "<Chariot Warrior216>"). Note that when this is done the first Warrior is "selected" on the map. In the Name text field, type "Chariot 1" to replace the default name. This Object has been renamed. Rename the other Warriors in the same way, calling them Chariot 2, Chariot 3, etc.
5. Next we need to create some Effects. Click the "Effect" button. Now click "New" to create a new Effect. The default Type Of Effect is "Object" and this is the one that is needed.
6. Click on the drop down list labeled "Action" (on the left) and select "Task-Area" from the list. The Task-Area action is used for moving an object to an Area on the map. In the first "Object" drop down list (furthest to the left), select Chariot 1. Chariot 1 is the Object that will move. In the top-right "Object" drop down list select Area 1. This is the target for the first Object; in other words, Chariot 1 is being commanded or "tasked" to move to Area 1.
7. Beneath the second "Object" drop down list, is another drop down list called "Action" which is a carryover from Empire Earth and has no effect. Leave this as "Normal" (which means the unit will just move to the target Area).
8. Now create three more Task-Area Effects in the same way, one for each remaining Chariot to its corresponding Area. For this example, leave the default names for all these Effects.

9. Next we need to create the Conditions for the Trigger. A Condition is used by the Trigger to determine if the Trigger should "fire" (i.e., play its Effects). To set up a new Condition, click the "Condition" button and then click "New." In the Type of Condition drop down list, select "Cheat Message Contains." In the text field, type in "move". This Condition will be true when the player types "move" into the cheat window during the scenario. As soon as the Condition is evaluated as true, the Trigger fires and the Chariots move to the Areas.
10. The name of this Condition could be changed, but for the purposes of this example leave the default name: <Cheat="move">.
11. Now it is time to assemble the entire Trigger. Click the "Trigger" button and click "New" to create a new Trigger. At the top-left of the lower portion of the screen is the Condition list for this Trigger—the "If" portion of the Trigger's "If-Then" Statement. Currently, the default "Always True" is selected. Click on the drop down list. You will see a number of default Conditions. Select the Cheat Message Contains Condition just created, which is called <Cheat="move">.
12. On the bottom left is the Effect list—the "Then" portion of the Trigger Statement. Click on the "Then" drop down list (currently the default "Do Nothing" is displayed) and choose the Effect: <Task-Area Chariot 1>. To add the other three Effects to this Trigger, move over to the drop down list that says "End" and select "And" to add another Effect. With the newly added Effect highlighted, click the "Then" drop down list again and select the next "Task-Area" Effect. Repeat this until all four of the Task-Area Effects have been added to this Trigger.
13. The Trigger is complete! Save the scenario, exit the Scenario Editor, and play the scenario to test it. Once the scenario is loaded, hit the Enter key to bring up the Cheat window, then type in the word "move" and press Enter again. The four Chariots move to their corresponding Areas, then decide to emigrate from the nome.

Appendix 2: Scenario Optimization

This Appendix is intended to offer advice on creating performance-friendly scenarios. Consider the variety of different computer systems out there. What kind of system might a potential player of the scenario have: top of the line or barely reaching the minimum system requirements? It is always safer to assume the latter and tailor scenarios appropriately so that they should run well on any system. Keep the following guidelines in mind and try, whenever possible, to test scenarios on more than one computer, especially if the scenario is intended for release to the world. A little forethought and careful design can prevent possible performance problems for players with slower computers, who should be able to enjoy the scenario just as much as players with cutting edge systems.

Possible Performance Hits

A performance hit is when something in a scenario, under certain conditions, causes a momentary pause or game lag. When something is said to be “expensive” it means it might cause a performance hit in some circumstances. The following are things to keep in mind when creating a scenario. Note that this doesn't mean these things shouldn't (or should) be done, rather that one should be mindful that they *could* result in a performance hit on some computers, under certain circumstances.

Game Wide

Map

- **Map Size** - Larger maps are more expensive than smaller maps. Keep maps as small as possible for the scenario. Smaller maps are also more difficult to build on, so this can increase scenario difficulty and challenge for the player.

Unit Count

- **Total Units** - For best results, try to keep the total number of units in the scenario below about 2700. This includes buildings, trees, and ambient objects like rocks, and resources.
- **Physics Units** - To help ensure good performance on even minimum systems, try to keep the total number of “physics units” below 600 at any given time. “Physics units” are those units that are mobile and currently on the map... even if they're standing still!
- **Unit Types** - The number of *types* of units should be minimized as well. Every additional type of unit (e.g., Chariot Warrior, Baker, Farmer, Nobleman, Priest, etc.) uses up more RAM.

Physics and Frame Rate

- **Pathing** - When units move from one location to another, their paths must be computed and this takes up CPU resources which can result in lag if there is no open path between the start point and the destination (such as if the units are blocked by a wall or cliffs or other impassable terrain).
- **Graphical Effects** - Creating multiple special effects like fire, smoke, explosions, etc. may cause a performance hit, especially when created en masse.
- **Large Scale Raider Invasions** - A large invasion with many raiders may slow down the scenario on low-end computers. Projectiles, different animations, damage effects, target selection and pathing, etc. all contribute to possible lag.

Trigger-Based Factors

- **Many Active Triggers** - Having lots of active (ON) Triggers can be expensive! Keep Triggers inactive (OFF) until they are needed, then have another Trigger turn them on.

- **Area-Based Computations** - The following Trigger tests can be expensive, especially if there are a lot of them:
 - Object in Area, especially for large areas or broadly defined Objects (see below)
- **Broad Object References** - Referenced (i.e., not selected on the map) Objects must scan every object that exists to identify the ones to use. It is best to narrowly define Object Specifications as much as possible. For example:
 - **Family Vs. Class** - Whenever possible, be specific about the desired kind of object. For example, it's better to indicate a specific unit type using "Class" rather than a more general type using "Unit Set." Also, avoid using the Not checkbox when specifying a Class or Family when possible as it is less expensive to say what object is sought than what is not.
- **Nested Objects** - Having Objects nested in other Objects can impact performance, especially if there are many such nested Objects and/or when used in conjunction with Object In Area.

Creating Smarter Triggers

The following are tips and tricks to keep in mind when creating Triggers and their components. For more tips, see the Optimizing Scenarios section, below.

- When spawning units on the map, set a unit attribute such as Wheat in the Object Specification with HAS ATTRIBUTE, so that another Effect that sets that attribute right afterward is not required.
- Use **non-dynamic** Objects whenever possible. By unchecking the Dynamic checkbox, the Object Specification is evaluated only once and keeps using the same unit(s), rather than evaluating the Specification multiple times.
- When creating multiple Task-Area Effects for the same units, queue the tasks together in one Trigger instead of separate ones. Create multiple Effects that task the Object to each Area in turn, and be sure to use the Delay box. This condenses multiple Triggers into one and saves the game from constantly evaluating Object in Area Conditions. For example:

If "Always True" then "Task Chariot to Area1" AND "Delay 30," "Task Chariot to Area2."

- Break up a complicated or potentially expensive move order into several moves. For example, imagine a large land mass that's walled off except for one gate. Instead of tasking units to an Area beyond the wall, task them up to the gate, through the gate, then to the final destination. This decreases one large pathfinding task into three much simpler ones.
- For Triggers with expensive Conditions that don't need to be on a "hair trigger" to fire, set the test interval higher than the default 500ms (game time). That way, they are evaluated less frequently.

General Optimization Tips

When it comes time to optimize a scenario, here are some things to try or at least keep in mind:

1. **Testing and Assessing Scenario Performance**
 - a. Play through the scenario (preferably on a low-end machine) and pay attention to lag to find any performance trouble spots.

- b. **Remember:** When playing the scenario, try all kinds of strange things - don't just "play it straight." Look not just for all possible performance issues, but also to make sure the scenario works as intended!
 - c. Use the tips listed earlier to step through every Trigger, making all the scenario and Trigger optimizations that are possible.
- 2. **Saving RAM with the Map, Units, and Physics**
 - a. Remove as many trees and ambient objects as possible.
 - b. Task all units that are not yet needed off map.
- 3. **Triggers**
 - a. Only necessary Triggers should be on!
 - i. **Remember:** every Trigger that is active is automatically evaluated over and over until it fires. This can cause a performance hit if there are a lot of Triggers in a scenario (~100 or more).
 - ii. As a general rule, all Triggers that don't need to be active for the entire scenario should be turned on only when they are needed, and turned off when they are not needed. *Especially looping Triggers!*
 - b. Looping Triggers should have delays!
 - i. Looping Triggers that have no delay for their Effects will fire "constantly," so it's best to include a delay time (the longer the better).
 - ii. If a Trigger is always true, or has a Condition that will often be true for a long period of time, setting "Conditions True For" to 1 second (or more) can improve performance.
- 4. **Conditions**
 - a. Conditions are often more expensive than Effects. Conditions, especially area-computation based Conditions, are evaluated constantly, whereas Effects are only evaluated each time their Trigger fires. So optimizing Conditions can have a much greater impact on improving scenario performance.
 - b. Optimize Condition Order
 - i. Remember that Trigger Conditions are evaluated in order.
 - ii. Whenever possible, place low-cost Conditions before (ahead of) expensive ones. Often this can turn an expensive Trigger into an inexpensive one.
 - iii. Set the Trigger Update Interval to a larger time interval for any necessary but expensive Triggers, whenever possible. (Remember the interval is rounded to the nearest half-second of game time.)
 - c. No qualifiers if not needed!
 - i. Unnecessary qualifiers may make an Effect not do anything, and it just adds evaluating time to a Condition.
 - ii. If the same Object is used as both the Condition and the Effect of a Trigger, check to see if the qualifier(s) can safely be removed.
 - iii. Examples:
 1. <create Chariot near Training Ground> won't create a Chariot if the Training Ground isn't there, so maybe there's no need to check.
 2. <task any Nobleman to any player building in area> if there's no player building in the area, the Noble won't go there. Either the player qualifier is unneeded or, if it is, perhaps the Area is not important.
 - d. Combine Multiple Triggers with the same Condition
 - i. Try this if there are many (say, 3 or more) Triggers that reference the same Condition (especially if they are looping Triggers).

- ii. Create one Trigger to evaluate an expensive Condition and then set a player variable to indicate whether that Condition is true.

Example:

Trigger1:

IF <Player Any Unit in Area> THEN <Set Player Variable1 = 1> AND <Turn Trigger2 ON>

Trigger2:

IF NOT <Player Any Unit in Area> THEN <Set Player Variable1 = 0> AND <Turn Trigger1 ON>

Trigger3, 4, etc.:

IF <Player1 Variable1 = 1> AND <other Conditions> THEN...

5. Effects

a. Graphic Effects

- i. Creating many graphic effects at once can cause a performance hit.
- ii. Minimize the number of graphic effects going off at once. Either reduce their number or try staggering them (for example, with delays) so that they do not all play simultaneously.

Index

- 'x' Description, 35, 36
- A.D. / B.C. Selector**, 31
- Acknowledgements, 4
- Active, 35, 36, 37, 38
- Add / Replace, 36, 38
- Aftermath**, 32
- Allow Incomplete, 27
- Amount, 33
- And**. *See* Operators
- Appendices, 43
- Area**, 13, 29
 - Center Camera On Area, 29
 - Description, 30
 - Select Rectangular Area, 29
- Background Image, 39
- Buildings**. *See* Construction
- Campaign Editor, 39
 - Add Button, 40
 - Available Scenarios, 42
 - Background Image, 39
 - Campaign Name, 39
 - Campaign Tier, 40
 - Campaign Tier Description, 40
 - City Name, 41
 - Clear, 41
 - Description, 40
 - Exit Button, 42
 - Icon, 42
 - Load Button, 42
 - New Button, 42
 - Remove Button, 41
 - Save Button, 42
 - Scenario Information, 41
 - Scenario Tier, 41
 - Scenarios in Campaign, 40
 - Starting the Campaign Editor, 39
 - Update Button, 41
 - XPos, 41
 - YPos, 41
- Campaign Name, 39
- Cancel, 36, 38
- Carryover, 35, 36, 37, 38
- Center Camera On Area, 29
- Change Text, 23
- Cheat Message Contains, 19
- City Patron Deity**, 30
- Class, 27
- Condition**, 12, 17
 - Cheat Message Contains, 19
 - Game-Attribute, 18
 - Object Exists, 17
 - Player Conditions, 17
 - Player Event Happened, 19
 - Random, 19
 - Trigger-Attribute, 18
- Conditions True For, 14
- Construction, 30
- Create Object, 20
- Current Pharaoh**, 31
- Deactive, 37
- Debrief Page, 32
- Defeat, 32
- Delay**, 33. *See* Operators
- Delete Goal, 34
- Description, 30
- Disabled Deities**, 30
- Disabled Events**, 30
- Distance To, 28
- Duration in Seconds, 33
- Dynamic, 27
- Effect**, 12, 19
 - Game, 24
 - Object, 20
 - Player, 21
 - Trigger, 22
 - World Site, 25
- Effect of Goal, 32
- Elevation Files, 6
 - Land Cutoff, 8
 - Land Height, 8
 - Water Cutoff, 8
 - Water Depth, 8
- Elevation Page
 - River Creation, 10
- Elevations Page, 8
 - Land Elevation, 9
- Enabled Deities**, 30
- Enabled Events**, 30
- End**. *See* Operators
- Events, 30
- Exit**, 6
- Exit Campaign Editor, 42
- Export**, 31
- Face Object, 21
- Family, 26
- Freeze Object, 20
- Game, 24
 - Graphic Effect, 24
 - Set Next Flood, 25
- General Optimisation Tips, 46
- Goal Category, 33
- Goal List, 32

- Goal Overview, 33
- Goals Page, 32
- Graphic, 36, 38
- Has Attribute, 29
- History**, 31
- Hot Key for Quick Load**, 5
- Import**, 31
- In Area, 28
- Inactive, 35, 37
- Include in Cinematic, 15
- Initial Prestige**, 31
- Inst. Sound Over**, 31, 32
- Is Closed, 19
- Is Dead, 19
- Is On, 18
- Is Open, 18
- Is Visible, 19
- Kill Object, 20
- Land Cutoff**, 8
- Land Elevation, 9
- Land Height**, 8
- Load**, 6
- Load Campaign, 42
- Loading, Saving, and Testing Scenarios, 5
- Localization Text File Name**, 31
- Looping, 15
- Map Label, 35, 37
- Map Page, 6
 - Elevation File, 6
- Media, 23
- Min and Max, 28
- Modifiers, 18
- Modify AI Military Target, 25
- Move, 36, 38
- Movie**, 31, 32
- Name, 32
- New Activation Site, 36
- New Campaign, 42
- New Scenario Creation, 35
- Not**. *See* Operators
- Notes to Scenario and Campaign Editors, 4
- Object**, 13, 26
 - Object Autonames, 26
 - Object Specification, 26
 - Object States, 28
 - Select On Map, 26
- Object Autonames, 26
- Object Exists, 17
- Object Specification, 26
- Object States, 28
- Operators, 15
 - And, 16
 - Delay, 16
 - End, 16
 - Not, 15
 - Or, 16
- Options, 30
- Or**. *See* Operators
- Our City, 36, 38
- Owner, 26
- Paint Roads, 11
- Play Sound, 23
- Player, 21
 - Award Victory, 21
 - Defeat, 21
 - Event, 22
 - Mini-map Flare, 22
 - Script Camera, 21
 - Set Attribute Level, 22
- Player Attribute for Goal, 33
- Player Conditions, 17
- Player Event Happened, 19
- Player Has Attribute, 17
- Players Page, 30
 - Construction, 30
 - Events, 30
 - Options, 30
 - Worship, 30
- Prestige Buildings**. *See* Construction
- Prop Build, 37
- Prop Descrip, 37
- Qualifier, 33
- Quick Load**, 5
- Quick Save**, 5
- Random, 19
- Remove Object, 20
- Remove Sprites, 11
- Replace Goal, 34
- Reset Some Entry Fields, 33
- Resources Area, 37
- River Creation, 10
- Save**, 5
- Save Campaign, 42
- Scenario Editor, 5
- Scenario File, 36
- Scenario Inst. Map**, 31
- Scenario Instructions**, 31
- Scenario Name**, 31
- Scenario Optimization, 45
- Scenario Performance Hits, 45
- Scenario Tips, 33
- Select On Map, 26
- Select Rectangular Area, 29
- Selected By, 29
- Send Chat Message, 24
- Send Dialogue, 23

- Starting Bricks**, 30
- Starting Food**, 31
- Starting Pharaoh Age**, 31
- Starting Pharaoh Number**, 31
- Starting the Scenario Editor, 5
- State. *See* Trigger
- Story Page, 31
- Submit as Goal, 34
- Task Area, 20
- Task Object, 20
- Terrain, 10
- Terrain Page, 10
 - Paint Roads, 11
 - Remove Sprites, 11
 - Terrain, 10
- Test**, 5
- Text Align Left, 36, 38
- The Trigger, 18
- Time Period**, 31
- Tips**, 31
- To create a new map**, 6
- To place objects on the map**, 11
- Trigger**, 12, 14
 - Creating Smarter Triggers, 46
 - Fired, 14, 18, 23
 - Looping, 15
 - Media, 23
 - Operators, 15
 - State, 22
- Trigger (Effect), 22
- Trigger Example, 43
- Trigger is On, 14
- Trigger Order, 14
- Trigger Statement, 15
- Trigger-Attribute, 18
 - Has Fired, 18
 - Is On, 18
 - The Trigger, 18
- Triggers Page, 12
 - Area, 13, 29
 - Component Names and "Autonames", 14
 - Condition, 12, 17
 - Deleting Conditions and Effects, 16
 - Effect, 12, 19
 - Inserting Conditions and Effects, 16
 - Object, 13, 26
 - Trigger, 12
 - Trigger Order, 14
 - Trigger Statement, 15
- Tutorial Mode Checkbox**, 6
- Unfreeze Object, 20
- Unit Attribute, 21
- Units**. *See* Construction
- Units Page, 11
 - Other Unit Controls, 12
 - To place objects on the map, 11
- Use Object, 27
- Using the Campaign Editor, 39
- Victory (AND), 33
- Victory (OR), 32
- Water Cutoff**, 8
- Water Depth**, 8
- Week**, 31
- World Level Page, 34
- World Map Selection, 34
- World Site, 18, 25
 - Close Site, 25
 - Is Closed, 19
 - Is Dead, 19
 - Is Open, 18
 - Is Visible, 19
 - Kill Site, 25
 - Make site invisible, 25
 - Make site visible, 25
 - Move Site, 25
 - Open Site, 25
 - Spawn Caravan, 25
- Worship, 30
- X Pos, Y Pos, 35, 37
- Year**, 31