New Effects and Conditions

New Effects

- **Change Rate of Fire:** Changes how much time a unit (or units) takes to reload its attack. Works in a similar way to the Change Object Speed effect (that is, any values typed in the “Quantity” field will be divided by 100, before being applied) and has no special modes.
- **Disable Advanced Buttons:** Disables the use of the Advanced Buttons. It has no special modes.
- **Change Armor Class:** Changes the armor value of a particular class for a unit (or units). In order to this work, “Number” needs to be set to the desired armor class ID and the target unit(s) need to have an armor entry set to this class in the data file.
- **Change Attack Class:** Changes the attack value of a particular class for a unit (or units). In order to this work, “Number” needs to be set to the desired attack class ID and the target unit(s) need to have an attack entry set to this class in the data file.
- **Change Default Armor:** Changes the Default Armor value for a unit (or units), which is used for all attack types which doesn’t have a matching armor type in the unit’s data. There are no special modes for this effect
- **Change Resource:** Changes the value of a particular resource for a given player. In order to this work, the “Number” field must be set to the ID of the Resource to be changed.
- **Change Object Resource:** Changes how much resources an object (or objects) is carrying. The ID of the carried resource can be changed by setting the value of the “Number” field to the ID of the desired resource plus one. In example, if you want to change the resource that an object is carrying to Food (Resource ID 0), you would need to set “Number” to 1.
- **Change Line of Sight:** Changes the Line of Sight value for a unit (or units). While using this effect, keep in mind that the maximum effective value for Line of Sight is 20 and any value above that will be treated as 20 by the game itself. There are no special modes for this effect.
- **Change Work Rate:** Changes the Work Rate value for a unit (or units). Works in a similar way to the Change Object Speed effect (that is, any values typed in the “Quantity” field will be divided by 100, before being applied) and has no special modes.
- **Change Hero Status:** Sets the value of the Hero flag of the selected units to the value of the “Number” field.
- **Change Icon ID:** Sets the icon ID of the selected unit(s) to the ID given in the “Number” field.
- **AI Script Goal Off:** Sets the status of the given AI Script goal to off.
- **Guard:** Orders the selected unit(s) to guard a unit or a building. Selection is limited to 40 units.
- **Follow:** Orders the selected unit(s) to follow a unit. Selection is limited to 40 units.

- **Scout:** Orders the selected unit(s) to scout around a given location. Selection is limited to 40 units.
- **Change Object Graphics:** Sets a particular graphic of the selected unit(s) to the ID given in the “Quantity” field.

Variable dependent effects

The following effects make use of the Variable system, that allows the use of 256 variables, which are reset every time a game starts and which can be read and manipulated by a set of conditions and effects. The effects that depend on the variable system can be found below:

- **Change Variable:** Changes the value associated to a variable. In order to this effect work, the value of the “Number” field needs to be changed to the ID of the variable that you want to change (which must be between 0 and 255). There are 3 modes available for this effect: Set, Add and Multiply

- **Reset All Variables:** Resets the value of all 256 variables to 0.
- **Change Resource by Variable:** Changes the value of a resource by the value of one of the 256 variables. In order to this effect work, the value of the “Quantity” field needs to be changed to the ID of the variable that you want to use (which must be between 0 and 255). There are 8 modes available for this effect:

    - **Set Resource Value by Variable**
    - **Add Variable Value to the Resource**
    - **Subtract Variable Value from the Resource**
    - **Multiply Resource Value by the Variable Value**
    - **Set Variable Value by Resource**
    - **Add Resource Value to Variable**
    - **Subtract Resource Value from Variable**
    - **Multiply Variable Value by Resource Value**

- **Save Variables to File:** Saves all of the 256 variables to a binary file, which can be read later. The file will be saved to the <AoK Folder>\Scenario folder (or to Games\<Your Mod Path>\Scenario folder, if the patch was applied over the .exe of a UserPatch based mod) and will bear the .bpvar extension. This effect does not work in Multiplayer games.
- **Load Variables from File:** Loads all of the 256 variables from a given file. The game will look for the file in the <AoK Folder>\Scenario folder (or to Games\<Your Mod Path>\Scenario folder, if the patch was applied over the .exe of a UserPatch based mod). This effect does not work in Multiplayer games.
- **Randomize Variable:** Sets the value of a variable to a random value between given minimum and maximum values. In order to this effect work, the value of

the “Number” field needs to be changed to the ID of the variable that you want to use (which must be between 0 and 255).

- **Pick Random Value:** Sets the value of a variable to a randomly chosen value among a given set of values. In order to this effect work, the “Message” field needs to contain the set of values, delimited by a space character, and the “Number” field needs to be changed to the ID of the variable that you want to use (which must be between 0 and 255). Parser directives can be used, as long as they only return integers, but not strings or float values.
- **Save Value to Variable:** Sets the value of a given variable to the current game time in seconds, years or to the coordinate of a selected tile. In order to this effect work, the value of the “Number” field needs to be changed to the ID of the variable that you want to use (which must be between 0 and 255).
- **Create Object by Variable:** Reads an unit ID from a given variable, and places that unit in the map tile denoted by the immediate successor of that variable. In order to this effect work, the value of the “Number” field needs to be changed to the ID of the variable that you want to use (which must be between 0 and 255).

Effects with string parsing

The following effects parse their string inputs before displaying them, by processing and replacing parser arguments:

- **Display Parsed Instructions**
- **Send Parsed Chat**
- **Parse Object Name**
- **Pick Random Value**

Parser arguments are enclosed by curly brackets and can be used to insert dynamic values like player names or resource values in a game message or object name.

The supported parser arguments are listed below:

- **{P:x}** - Displays the name of the player whose ID is *x*
- **{R:p:x}** - Displays, as an integer, the value of resource ID *x* of the player whose ID is *p*
- **{Rf:p:x}** - Displays, as floating point number with 2 decimal digits, the value of resource ID *x* of the player whose ID is *p*
- **{C:p}** - Displays the name of the civilization of the player whose ID is *p*
- **{S:x}** - Display the text of the String ID *x*
- **{V:x}** - Displays the value of the Variable ID *x*
- **{T}** - Displays current game time in the *HH:MM:SS* fomat
- **{Th}** - Displays the current game time in the *HHhMMmSSs* format
- **{T:s:e}** - Displays the current game time in years relative to a starting point, defined by *s* and an ending point defined by *e*. If the ending point is not set to a value greater than 0, only the starting point will be taken into consideration.

Variables can be used as parser argument parameters by placing *v* before the parameter value, followed by the variable ID from which the parameter value should be obtained.

The *Display Parsed Instructions* and *Parse Object Name* effects support the *up_effect* and *up_attribute* modes, respectively, added in UserPatch 1.5, thus allowing to replace the parameters used in those modes by dynamic values like resource or variable values, through the use of parser arguments.

New Conditions

- **Own Fewer Foundations:** Counts the number of building foundations a player has in a particular area or in the entire map. This condition becomes true if the amount of foundations counted is less than or equal to the amount given.
- **Own Fewer Rubble:** Counts the number of destroyed buildings a player has in a particular area or in the entire map. This condition becomes true if the amount of destroyed buildings counted is less than or equal to the amount given and doesn't count previously placed rubble.
- **Room Setting:** Checks a particular room (or game) setting and becomes true if the given room setting is set. The room setting to be checked can be chosen by setting the value of the "Room Setting" drop-down box to one of the following values:
  - **Multiplayer Game**
  - **Teams Together**
  - **Lock Teams**
  - **Lock Speed**
  - **Allow Cheats**
  - **Starting Resources**
    Note: In order to check this setting, "Quantity" must be set to the amount referent to the desired resource setting:
    0. **Standard**
    1. **Low**
    2. **Medium**
    3. **High**
  - **Population**
    Note: In order to check this setting, "Quantity" needs to be set to the Population Limit you want to check.
  - **Game Speed**
    Note: In order to check this setting, "Quantity" needs to be set to the desired speed amount multiplied by 10. So, if you want to check if the Game Speed was set to Normal (1.5), then you need to set the "Quantity" field to 15.
  - **Reveal Map**
    Note: In order to check this setting, "Quantity" must be set to the amount referent to the desired reveal map setting:
    0. **Normal**
    1. **Explored**

2. **All Visible**
3. **No Fog**

- **Victory**
  Notes: In order to check this setting, “Quantity” must be set to the amount referent to the desired victory condition setting. Time Limit and Score are checked in separate modes and this condition checks the victory set in the lobby, instead of the one set in the scenario
  0. **Standard**
  1. **Conquest**
  2. **Relics**
  3. **Last Man Standing**
- **Time Limit Victory**
  Note: In order to check this setting, “Quantity” needs to be set to the desired Time Limit amount for victory.
- **Score Victory**
  Note: In order to check this setting, “Quantity” needs to be set to the desired Score amount for victory.
- **All Techs**

- **Trigger Active:** Checks whether a trigger is active or not. Becomes true if the given trigger is active.
- **Computer Player:** Checks whether a selected player is a computer player or not. Becomes true if the given player is a computer player (i.e. if it’s being played by an AI).
- **Variable Value:** Checks the value of a given variable ID and becomes true if this value matches the value given in the “Quantity” field.
- **Variable Value Lower Than:** Checks the value of a given variable ID and becomes true if this value is lesser than the value given in the “Quantity” field.
- **Technology State:** Checks the state of a technology and becomes true if the state of the technology matches the given state.

New Unit Attributes for Technologies

The attributes listed below can be used in both *.dat* technologies and in the *up-effect*/*up-attribute* modes. It’s worth to mention that those new attributes can only be used in the “Set Attribute” technology effect mode.

- **58**: Unit Icon
- **59**: Resource Storage Mode
  - Note: Value has to be set to <*resource ID*>*64 + <*storage mode ID*>, in order to change the resource storage mode for a particular unit resource.
- **60**: Language DLL Hotkey Text
- **61**: Special/Charging Graphic (Types 70+ only)
- **62**: Special/Charging Mode (Types 70+ only)
- **63**: Drop Site #1 (Types 70+ only)

- **64**: Drop Site #2 (Types 70+ only)
- **65**: Villager/Unit Swap Mode (Types 70+ only)
- **66**: Initiates Research (Type 80 only)
- **67**: Transform Unit (Type 80 only)