

Restrictions

Edit

New page

friarsol edited this page on Apr 10, 2022 · 1 revision

Activation

Activation\$ can accept several different values:

- Metalcraft
- Threshold
- Hellbent

These basically follow the text of the card.

ActivationZone

ActivationZone\$ For cards that have abilities that you can activate outside of the Battlefield

Flashback

While Flashback isn't technically an SA_Restriction it's listed here because it's handled very similarly in AFs. If sets the ActivationZone to Graveyard and sets the FlashbackAbility flag of a SpellAbility which will exile the card as it finishes resolving.

SorcerySpeed

SorcerySpeed \$True For cards that can only activate if you could cast a sorcery.

PlayerTurn

PlayerTurn \$True For cards that can only be activated during the activating player's turn.

OpponentTurn

OpponentTurn \$True For cards that can only be activated during the opponent's turn of the activating player.

AnyPlayer

AnyPlayer \$True For cards that can be activated by any player. Prophecy has lots of examples of these types of cards.

► Pages 52

Forge Wiki Home



- [User Guide](#)
 - [FAQ](#)
 - [Troubleshooting](#)
 - [AI](#)
 - [Network Play](#)
 - [Network FAQ](#)
 - [Network Extra](#)
- [Adventure Mode](#)
 - [Gameplay Guide](#)
 - [Getting Started](#)
 - [Towns & Capitals](#)
 - [Dungeons](#)
 - [Equipments and Items](#)
 - [Modding and Development:](#)
 - [Create Enemies](#)
 - [Create Rewards](#)
 - [Create Maps](#)
 - [Configure Planes](#)
- [Card Scripting API](#)
 - [Ability effects](#)
 - [Triggers](#)
 - [Replacements](#)
 - [Costs](#)
 - [Affected / Targets](#)
 - [Restrictions](#)
- [Development](#)
 - [IntelliJ Setup](#)
 - [Android Builds](#)
 - [Ownership](#)
 - [Simulation AI](#)
- [Customization and Themes](#)
 - Skins

ActivationLimit

ActivationLimit\$ For cards that have a limited amount of uses

ActivationNumberSacrifice

ActivationNumberSacrifice\$ For cards that if they you activate them more than a certain amount per turn are sacrificed at end of the turn.

ActivationPhases

ActivationPhases\$<Phases,Seperated,By,Commas> For Abilities can only be activated during upkeep then it'll be ActivationPhases\$Upkeep if it's before Declare Attackers than it's
ActivationPhases\$Upkeep,Draw,Main1,BeginCombat

This can also be handled in a range. ActivationPhases\$ BeginCombat->EndCombat for Spells that can only be cast during combat.

ActivationCardsInHand

ActivationCardsInHand\$ For Abilities with Hellbent (for 0) or Library of Alexandria (for 7)

Planeswalker

Planeswalker\$ True For Planeswalker Abilities. These can only be activated at Sorcery Speed, and only if no Planeswalker Abilities (including this ability) on this card have been activated that turn.

Add Ultimate\$ True when applicable for the AI and achievements.

Present

Present is a Restriction that comes in two parts. IsPresent is required. And PresentCompare is optional.

IsPresent

IsPresent\$ gets all cards on the battlefield that is considered a ValidCard.

PresentCompare

PresentCompare\$<2 Letter Comparison>. The comparisons are LE,LT,EQ,NE,GE,GT and translated to <=, <, ==, !=, >=, >. The comparator is on the right side of the equation.

If PresentCompare is **missing**, the Restriction defaults to GE1.

Condition

- Music & Sounds
- [Card Images](#)
- [File Formats](#)

- [Missing Cards in Forge](#)
- [Credit and Thanks](#)

Clone this wiki locally

<https://github.com/Card-Forge>



Condition is similar to a IsPresent restriction, except it's checked on Resolution of the spell and not on Activation.

CheckSVar

CheckSVar specifies that the results computed from an SVar (Usually via xCount) must equal a certain value which you can specify in the accompanying SVarCompare parameter. SVarCompare is the same as PresentCompare above.

+ Add a custom footer

It means, "the controller of the enchanted thing." This would be the "that player" part of things like "At the beginning of enchanted creature's controller's upkeep, that player loses 1 life."

EnchantedOwner

AttackingPlayer

Things like "Whenever Souls of the Faultless is dealt combat damage, you gain that much life and attacking player loses that much life."

DefendingPlayer

Things like "Whenever CARDNAME deals combat damage, defending player discards a card."

Player

Each player.

Defined Cards

Defined Cards are for SAs like Pump or Regenerate. They tell you who is receiving this action (but not in a targeted way).

Self

The most common of these is Defined\$ Self. It means exactly what you think "I gain Flying" "Regenerate Me." Abilities that don't seem to define anything should affect Cards default to "Self"

AB\$Regenerate| Defined\$ Self

It is important to include this in SAs that have an unclear "Default" value such as Damage.

Enchanted

Enchanted is fairly common. It means "do this action to the card I'm Enchanted to"

A simple example is a card that Pumps the attack of the Enchanted creature

AB\$Pump | Defined\$ Enchanted | NumAtt\$ +1

Equipped

Equipped isn't very common, but is very similar. It means "do this action to the card I'm Equipped to"

AB\$Pump | Defined\$ Equipped | NumAtt\$ +2 | NumDef\$ +2

- Music & Sounds
- [Card Images](#)
- [File Formats](#)

- [Missing Cards in Forge](#)
- [Credit and Thanks](#)

Clone this wiki locally

<https://github.com/Card-Forge>



Remembered

ThisTurnEntered

ThisTurnEntered lets you make the ability act on valid cards that were put in a certain zone, from a certain zone, this turn. The format is "ThisTurnEntered [from]".

For example: No Rest for the Wicked and the like would use: AB\$ ChangeZone | Cost\$ Sac<1/CARDNAME> | Defined\$ ThisTurnEntered Graveyard from Battlefield Creature.YouCtrl | SpellDescription\$ Return to your hand all creature cards in your graveyard that were put there from the battlefield this turn.

Targeted

Targeted will only appear on a [SubAbility][]. It means "Do this action to whatever a parent Ability targeted"

That may sound confusing so here's an example.

If you had a spell that says "Untap target Creature. It gains +1/+1 until end of turn" it would look like similar to this.

SP\$Untap | ValidTgts\$ Creature | SubAbility\$ DBPump

SVar:DBPump:DB\$Pump | Defined\$ Targeted | NumAtt\$ +1 | NumDef\$ +1

Targets

Each element follows the form:

CardType{.restriction}{+furtherRestriction}

The restrictions are optional.

CardType may be any type, generally the supertypes like Creature, Artifact, Enchantment, Land, etc. However, it could also be Elf or Goblin... though that would also include Elf Enchantments, for example. To specify an Elf Creature, then it should be "Creature.Elf". "Permanent" represents any permanent, "Card" any card....

Restrictions other than type that are interpreted: (case sensitive)

```
named{Name}
notNamed{Name}
sameName
NamedCard
```

Color, nonColor (White, nonWhite, etc)
 Colorless, nonColorless
 Multicolor, nonMulticolor
 MonoColor, nonMonoColor
 ChosenColor
 SharesColorWith

 YouCtrl, YouDontCtrl
 YouOwn, YouDontOwn
 EnchantedPlayerCtrl
 OwnerDoesntControl
 ControllerControls{type}
 RememberedPlayerCtrl (use if Remember will be set before resolution)
 TargetedPlayerCtrl (only use if restriction is needed while initial targeting happens)

 Other,Self

 AttachedBy,Attached
 DamagedBy,Damaged

 with{Keyword}, without{Keyword}
 tapped, untapped
 faceDown
 enchanted, unenchanted, enchanting, EnchantedBy
 equipped, unequipped, equipping, EquippedBy
 dealtDamageToYouThisTurn
 wasDealtDamageThisTurn
 wasDealtDamageByHostThisTurn
 wasDealtDamageByEquipeeThisTurn
 token, nonToken
 kicked, notkicked
 enteredBattlefieldThisTurn

 power{cmp}{X or #}
 toughness{cmp}{X or #}
 cmc{cmp}{X or #}
 {cmp} is a comparator:
 LT - Less Than
 LE - Less than or Equal
 EQ - Equal
 GE - Greater than or Equal
 GT - Greater Than
 X is parsed from "SVar:X:Count\$____"

 greatestPower, leastPower
 greatestCMC

 counters{cmp}{X or #}{Type}

attacking, notattacking
blocking, notblocking
blocked, unblocked, blockedBySource
kicked, notkicked
evoked
hasDevoured, hasNotDevoured
non{Type}
ChosenType{Type}
{Type}

CostsPhyrexianMana

Above, DirectlyAbove
TopGraveyardCreature
TopGraveyard

Cloned

isRemembered

Examples:

"Artifact or Enchantment" would be represented as 2 elements
"Artifact,Enchantment"

"non-black, non-artifact creature" would be represented as
"Creature.nonBlack+nonArtifact"

"creature with power 2 or less" would be "Creature.powerLE2"

"nonbasic land" is "Land.nonBasic"

"creature with flying" is "Creature.withFlying"

"creature self with four or more level counters" is
"Creature.countersGE4LEVEL+Self"

+ Add a custom footer

friarsol edited this page on Apr 10, 2022 · 1 revision

Cost is a class that attempts to streamline costs throughout all cards. It requires that each cost is separated by a space. I will use examples that could be found in Ability, although certain Keyworded abilities do use Cost too.

Common

Description

Description is an optional last parameter in the cost. This is to allow for complex Type definitions to have a nice Description that is readable.

CostDesc / PrecostDesc

UnlessCost

UnlessCost allows the player specified with UnlessPayer (same as Defined, defaults to TargetedController) to pay mana to prevent the resolving of the ability. If the script has the param "UnlessSwitched", then the player pays mana to resolve the ability (usually used to handle "any player may pay ...").

XChoice

XChoice is the variable that basically means "You can choose whatever you want for this variable. But you need to decide what X is before you start paying." This would commonly appear as an SVar definition of X.

xPaid

xPaid is the amount of Mana Paid for an X Cost. There are a few cards that will use the X Payment to determine other costs (like Abandon Hope) This would commonly appear as an SVar definition of X.

Pages 52

Forge Wiki Home

- User Guide
 - FAQ
 - Troubleshooting
 - AI
 - Network Play
 - Network FAQ
 - Network Extra
- Adventure Mode
 - Gameplay Guide
 - Getting Started
 - Towns & Capitals
 - Dungeons
 - Equipments and Items
 - Modding and Development:
 - Create Enemies
 - Create Rewards
 - Create Maps
 - Configure Planes
- Card Scripting API
 - Ability effects
 - Triggers
 - Replacements
 - Costs
 - Affected / Targets
 - Restrictions
- Development
 - IntelliJ Setup
 - Android Builds
 - Ownership
 - Simulation AI
- Customization and Themes
 - Skins

CARDNAME

For Costs that do something to themselves (ex. Discard Self, Sacrifice Self)

Types of Cost

Discard

Discard has two required parameters and one optional in the form
Discard<Num/Type/Description>

- The first is how many cards are being discarded.
- The second is what card types can be discarded. (Hand for the whole hand, Random for chosen randomly)

Draw

Exert

Exile

Exile has two required parameters and one option in the form of
Exile<Num/Type/Description>

There are also a few sister abilities that all fit under the Exile umbrella.

- Exile (for cards on the Battlefield)
- ExileFromGraveyard
- ExileFromHand
- ExileFromTop (for cards on top of your library, this doesn't default Type to Card, so make sure you add it)

Some Examples

- Exile<1/Creature>
- Exile<1/CARDNAME>
- ExileFromHand<1/CARDNAME>
- ExileFromHand<2/Creature>
- ExileFromGrave<1/CARDNAME>
- ExileFromGrave<1/Treefolk>
- ExileFromTop<10/Card>

- Music & Sounds

- [Card Images](#)
- [File Formats](#)

- [Missing Cards in Forge](#)
- [Credit and Thanks](#)

Clone this wiki locally

<https://github.com/Card-Forge>



FlipCoin

Only used by "Karplusan Minotaur".

Mana

- Cost\$ 2
 - 2 colorless mana
- Cost\$ B R
 - 1 black and 1 red mana
- Cost\$ WG
 - Hybrid White/Green mana
- Cost\$ S
 - Snow Mana
- Cost\$ Mana<2\Creature>
 - 2 colorless produced by a source with type 'creature'. Note the backslash - it was chosen because hybrid costs already use slash

Here's some examples:

- Discard<1/Card>
 - "Discard 1 Card"
- Discard<0/Hand> (The number is ignored when Hand is used as a type.)
 - Discard your hand
- Discard<2/Random>
 - Discard 2 Cards at Random
- Discard<1/CARDNAME>
 - Discard Self (CARDNAME)
- Discard<1/Creature.Black/Black Creature>
 - Discard 1 Black Creature

Mill

Subtract(Remove) Counter

SubCounter has two required parameters in the form of
SubCounter<Num/CounterName>

- SubCounter<2/P1P1>
- SubCounter<1/CHARGE>

Remember the token name should appear all in caps.

As third parameter you can use a ValidCard.

Sacrifice

Sacrifice has two required parameters and one optional parameter in the form of Sac<Num/Type/Description>

- Sac<1/Artifact>
- Sac<1/CARDNAME>

Tap

- Cost\$ T

Untap

- Cost\$ Untap

- or -

- Cost\$ Q

Unattach

PayEnergy

PayLife

PayLife has one required parameter in the form of PayLife

- PayLife<2>

GainLife

TapXType

TapXType has two required parameters and one option in the form of tapXType<Num/Type/Description>

- tapXType<3/Creature.White>

Return

Return has two required parameters and one optional in the form of Return<Num/Type/Description>

- Return<1/Land>
- Return<1/CARDNAME>

Putting it Together

Putting it together is pretty simple. If a card needs to pay mana and tap, it would look like this:

- Cost\$1 W T

For a spell that has an additional cost of sacrificing a land, put the mana cost and the additional cost in the cost:

- Cost\$2 G Sac<1/Land>

One of the features of Cost is you can have more than one of the same Cost type:

- Cost\$ Sac<1/Swamp> Sac<1/Creature>

There are many examples, but they mostly fall into those categories.

+ Add a custom footer

Triggers

Edit

New page

friarsol edited this page on Apr 10, 2022 · 1 revision

Triggers define when an ability should be automatically added to the stack. Their syntax in the card files are very similar to that used by the various AbilityFactorys, that is: every trigger line starts with "T:" (without the quotes) and is a collection of name-value pairs (name and value are separated by \$) separated by pipes (|). Like with AbilityFactorys, there are a few that show up almost always if not always. There are two parameters that are always present on a trigger:

- Mode - Specifies what kind of situation the trigger should wait for.
- TriggerDescription - Describes the trigger, much like AF's SpellDescription parameter.
- (Execute - Specifies the name of the SVar that holds the ability to run when the trigger goes off.)

NOTE: The Execute parameter may be absent, if you are writing a delayed triggered ability. See below.

Depending on which Mode is specified, other parameters may be expected. Below are the currently available modes.

The script has access to many things that were previously internal to triggers. These things are accessed via Triggered-variables. Triggered variables are always of the form "Triggered[Controller/Owner]" and are specific to each trigger mode. You can use Triggered-variables that return a card or a player directly in Defined\$ parameters or to grab extra info from (like you use "Targeted" for, for instance "SVar:X:TriggeredCard\$CardPower"). You can get the controller or owner of a card returned by a Triggered-variable by appending "Controller" or "Owner" to the variable. Triggered-variables that return an integer can only be accessed from Count\$, i.e. "SVar:X:Count\$TriggeredLifeAmount".

Other parameters that triggers can use are:

- Secondary - If a trigger has Secondary\$ True set, it means that it's trigger description won't show up in a card's text box. This can be used if you need to use several triggers to emulate a single effect on a real card.
- Static - This parameter is mainly used for "As CARDNAME enters the battlefield..." type things. It causes the triggered ability to resolve right away, instead of going on the stack.

▶ Pages 52

Forge Wiki Home



- [User Guide](#)
 - [FAQ](#)
 - [Troubleshooting](#)
 - [AI](#)
 - [Network Play](#)
 - [Network FAQ](#)
 - [Network Extra](#)
- [Adventure Mode](#)
 - [Gameplay Guide](#)
 - [Getting Started](#)
 - [Towns & Capitals](#)
 - [Dungeons](#)
 - [Equipments and Items](#)
 - [Modding and Development:](#)
 - [Create Enemies](#)
 - [Create Rewards](#)
 - [Create Maps](#)
 - [Configure Planes](#)
- [Card Scripting API](#)
 - [Ability effects](#)
 - [Triggers](#)
 - [Replacements](#)
 - [Costs](#)
 - [Affected / Targets](#)
 - [Restrictions](#)
- [Development](#)
 - [IntelliJ Setup](#)
 - [Android Builds](#)
 - [Ownership](#)
 - [Simulation AI](#)
- [Customization and Themes](#)
 - [Skins](#)

- ResolvingCheck
- NoResolvingCheck - makes a trigger not recheck its condition to resolve

Always

Always-triggers represent State triggers, a special kind of triggers. These triggers will not do any checks for intervening if-clauses and will not go on the stack if an instance of it has already been put there. They are checked every time state effects are checked.

Examples: Emperor Crocodile.

There are no special parameters and no triggered-variables.

Attached

Attached-triggers go off when a card is attached, via enchanting or equipping, to another card.

Examples: Bramble Elemental, Kithkin Armor.

There are 2 special parameters:

- ValidSource - The card that is being attached to another must match this for the trigger to go off.
- ValidTarget - The card that is having another card attached to it must match this.

There are 2 Triggered-variables:

- Source - The card that is being attached.
- Target - The card that is being attached to.

AttackerBlocked

AttackerBlocked-triggers go off when a creature becomes blocked. It goes off only once (no matter how many blockers there are) right after the declare blockers step.

Examples: AEther Membrane, Alley Grifters.

There are 2 special parameters:

- ValidCard - The attacking creature must match this for the trigger to go off.
- ValidBlocker - The blocking creature must match this for the trigger to go off.

There are 3 Triggered-variables:

- Attacker - The card of the attacking creature.
- Blocker - The card of the blocking creature.
- NumBlockers - The number of things blocking the attacker

- Music & Sounds

- [Card Images](#)
- [File Formats](#)

- [Missing Cards in Forge](#)

- [Credit and Thanks](#)

Clone this wiki locally

<https://github.com/Card-Forge>



AttackerUnblocked

AttackerUnblocked-triggers go off when a creature attacks and is not blocked, right after the declare blockers step.

Examples: Abyssal Nightstalker, Dauthi Mindripper

There is 1 special parameter:

- ValidCard - The attacking creature must match this for the trigger to go off.

There is 1 Triggered-variable:

- Attacker - The card of the attacking creature.

AttackersDeclared

Goes off after attackers are declared, if any attackers were declared, once a combat only.

Examples: Lightmine Field, Curse of Inertia. There are 2 special parameters:

- AttackingPlayer - The attacking player must match this.
- AttackedTarget - One of the game entities in TriggeredAttackedTarget must match this.

There are 3 Triggered-variables:

- Attackers - Collection of attacking creatures.
- AttackingPlayer - The targeted object.
- AttackedTarget - Collection of game entities that each attacker is attacking.

Attacks

Attacks-triggers go off when a creature attacks. That is, it goes off once for each creature that attacks during your each combat phase (Right after the declare attackers step).

Examples: Accorder Paladin, Trepanation Blade.

There are 2 special parameters:

- ValidCard - The attacking creature must match this for the trigger to go off.
- Alone - If this is True, the trigger will only go off if the creature attacks alone.

There is 1 Triggered-variable:

- Attacker - The card of the attacking creature.

BecomeMonstrous

BecomeMonstrous-triggers go off when a creature becomes Monstrous, naturally.

Examples: Arbor Colossus, Ember Swallow.

There is 1 special parameter:

- ValidCard - The card that becomes monstrous must match this.

There is 1 Triggered-variable:

- Card - The card that became monstrous.

BecomesTarget

BecomesTarget-triggers go off when a spell or ability (or either) is put on the stack targeting something that matches a Valid-expression.

Examples: Angelic Protector, Cephalid Illusionist.

There are 3 special parameters:

- SourceType - Can be Spell or Ability. It is optional.
- ValidSource - The card that targets something must match this.
- ValidTarget - The targeted object must match this.

There are 2 Triggered-variables:

- Source - The targeting card.
- Target - The targeted object.

BlockersDeclared

Goes off after blockers are declared if there are any, once in a combat only. (unlike Blocks, which goes off once for each creature that blocks.)

Examples: Tide of War

There are no special parameters.

There are 2 Triggered-variables:

- Blockers - Collection of all blockers.
- Attackers - Collection of all attackers.

Blocks

Blocks-triggers go off when a creature blocks (No surprise there ;)).

Examples: Amphibious Kavu, Cinder Wall (See below).

There are 2 special parameters:

- ValidCard - The blocking creature must match this.
- ValidBlocked - The creature being blocked must match this.

There are 2 Triggered-variables:

- Attacker - The card of the attacking creature.
- Blocker - The card of the blocking creature.

Championed

Goes off when a creature is championed.

Examples: Mistbind Clique.

There are 2 special parameters:

- ValidCard - The card being exiled for championing must match this.
- ValidSource - The champion card that is being played must match this.

There are 2 Triggered-variables:

- Championed - The champion-exiled card
- Card - The championing card.

ChangesController

Goes off when a card changes controller.

Examples: Coffin Queen, Duplicity.

There are 2 special parameters:

- ValidCard - The card whose controller changes must match this.
- ValidOriginalController - The player who originally controlled the card must match this.

There is 1 Triggered-variable:

- Card - The card whose controller changes.

ChangesZone

ChangesZone-triggers go off whenever, you guessed it, a card changes zone.

Examples: Contagion Clasp, see below.

There are 3 special parameters:

- ValidCard - The card that was moved must match this for the trigger to go off.
- Origin - The card must be moved from this zone for the trigger to go off. (Defaults to "Any")
- Destination - The card must be moved to this zone for the trigger to go off. (Defaults to "Any")

There is 1 Triggered-variable:

- Card - The card that was moved.

Clashed

Clashed-triggers go off whenever a player has clashed, regardless of whether you won or not. They are always run after the clash is through.

Examples: Entangling Trap, Rebellion of the Flamekin.

There are 2 special parameters:

- ValidPlayer - Who clashed.
- Won - True if the player must have won, false otherwise.

There are no Triggered-variables.

CombatDamageDoneOnce

Goes off once for every game entity that receives damage in combat.

Examples: Nature's Will, Pyrewild Shaman.

There are 2 special parameters:

- ValidSource - One or more of the game entities that dealt the damage must match this.
- ValidTarget - The game entity that received damage must match this.

There are 2 Triggered-variables:

- Sources - A collection of the game entities that dealt the damage.
- Target - The game entity that received damage.

CounterAdded & CounterRemoved

These triggers go off when a counter is added to / removed from a card.

Examples: Bloodcrazed Hoplite, Aeon Chronicler.

There are 2 or 3 special parameters:

- ValidCard - The card getting the counter must match this for the trigger to go off.
- CounterType - The counter must be of this type for the trigger to go off.
- NewCounterAmount - The counter amount AFTER the trigger fires must be this. NOTE: only available to CounterRemoved at the moment.

There is 1 Triggered-variable:

- Card - the card the counter was added to

Countered

Goes off when a spell or ability is countered.

Examples: Lullmage Mentor, Multani's Presence.

There are 3 special parameters:

- ValidCard - The host card of the spell/ability that was countered must match this.
- ValidPlayer - The player that cast/activated the spell/ability that was countered must match this.
- ValidCause - The host card of the spell/ability that did the countering must match this.

There are 3 Triggered-variables:

- Card - The host card of the spell/ability that was countered.
- Player - The player that cast/activated the spell/ability that was countered.
- Cause - The host card of the spell/ability that did the countering.

Cycled

Cycled-triggers simply go off whenever you cycle a card.

Examples: Astral Slide, Bant Sojourners.

There is 1 special parameters:

- ValidCard - The card that was cycled must match this.

There is 1 Triggered-variable:

- Card - The card that was cycled.

DamageDone

DamageDone-triggers go off whenever any source deals damage to any target.

Examples: Abomination of Gudul, Prophetic Flamespeaker.

There are 4 special parameters:

- ValidSource - The source of the damage must match this.
- ValidTarget - The target of the damage must match this.
- CombatDamage - If set to true, the trigger will only go off if the damage dealt was combat damage. If set to false, it will only go off if it wasn't combat damage.. If omitted, it will go off either way.
- DamageAmount - Specifies how much damage must be done for the trigger to go off. It takes the form "" where can be LT (Less Than), LE (Less than or Equal), EQ (Equal), GE (Greater than or Equal) or GT (Greater Than) and is a positive integer.

There are 3 Triggered-variables:

- Source - The source of the damage.
- Target - The target of the damage.
- DamageAmount - The amount of damage dealt(integer).

BEWARE: Since the Target variable can be either a player or a card, you must take care to limit this in the ValidTarget parameter!

DealtCombatDamageOnce

Goes off *once* for each creature that deals damage in combat.

Examples: Arashin War Beast, Five Alarm Fire.

There are 2 special parameters:

- ValidTarget - The game entity that receives damage must match this.
- ValidSource - The card that dealt damage must match this.

There are 2 Triggered-variables:

- Target - The game entity that receives damage.
- Source - The card that dealt damage.

Destroyed

Goes off when a permanent is destroyed. (Not if it is destroyed, but then regenerated.)

Examples: Cobra Trap, Sacred Ground.

There are 2 special parameters:

- ValidCard - The card that was destroyed must match this.
- ValidCauser - The player that activated the spell/ability that destroyed the card must match this.

There are 2 Triggered-variables:

- Card - The card that was destroyed.
- Causer - The player that activated the spell/ability that destroyed the card.

Devoured

Goes off when a creature is sacrificed for a Devour creature.

Examples: Kresh the Bloodbraided Avatar.

There is 1 special parameter:

- ValidDevoured - The devoured creature must match this.

There is 1 Triggered-variable:

- Devoured - The devoured creature.

Discarded

Discarded-triggers go off whenever a card is discarded from a players hand. (side note: ChangesZone-triggers may also go off here because the card is moved from the hand to the graveyard)

Examples: Abyssal Nocturnus, Confessor.

There are 3 special parameters:

- ValidPlayer - The player who discarded the card must match this.
- ValidCard - The discarded card must match this.
- ValidCause - The card that caused the player to discard must match this.

There is 1 Triggered-variable:

- Card - The card that was discarded.

Drawn

Drawn-triggers go off when a player draws a card.

Examples: Booby Trap, Kederekt Parasite.

There is 1 special parameter:

- ValidCard - The drawn card must match this.

There are 2 Triggered-variable:

- Card - The card that was drawn.
- Player - The player that drew the card.

Evolved

Goes off when a creature gets a +1/+1 counter from evolving.

Examples: Renegade Krasis.

There is 1 special parameter:

- ValidCard - The card that got the counter must match this.

There is 1 Triggered-variable:

- Card - The card that got the counter.

FlippedCoin

Goes off when a player flips a coin.

Examples: Chance Encounter, Karplusan Minotaur.

There are 2 special parameter:

- ValidPlayer - The player who flipped the coin must match this.
- ValidResult - If this parameter is "Win", the player must win the flip.

There is 1 Triggered-variable:

- Player - The player who flipped the coin.

LandPlayed

LandPlayed-triggers of course go off when a land is played.

Examples: Burgeoning, City of Traitors.

There is 1 special parameter:

- ValidCard - The played card must match this.

There is 1 Triggered-variable:

- Card - The card that was played.

LifeGained & LifeLost

These triggers go off on when a player either gains or loses life, of course.

Examples: Cradle of Vitality, Exquisite Blood.

There is 1 special parameter:

- ValidPlayer - The player who gained or lost life must match this.

There are 2 Triggered-variables:

- Player - The player that gained/lost life.
- LifeAmount - The amount of life lost/gained. (integer)

LosesGame

Goes off when a player loses the game.

Examples: Blood Tyrant, Elbrus, the Binding Blade // Withengar Unbound

There is 1 special parameter:

- ValidPlayer - The player who lost the game must match this.

There is 1 Triggered-variable:

- Player - The Player who lost the game.

NewGame

Goes off once at the start of each game, after mulligans.

Examples: Maralen of the Mornsong Avatar, Worldknit.

There are no special parameters or Triggered-variables.

PayCumulativeUpkeep

Goes off when a player pays or doesn't pay the cumulative upkeep for a card.

Examples: Balduvian Fallen, Heart of Bogardan.

There are 2 special parameters:

- Paid - Whether or not the player must have paid the cumulative upkeep.
- ValidCard - The card that has the cumulative upkeep must match this.

There are 2 Triggered-variables:

- Card - The card that has the cumulative upkeep.
- PayingMana - A string representing the mana spent.

PayEcho

Goes off when a player pays or doesn't pay the echo cost for a card.

Examples: Shah of Naar Isle.

There are 2 special parameters:

- Paid - Whether or not the player must have paid the echo.
- ValidCard - The card that has the echo must match this.

There is 1 Triggered-variable:

- Card - The card that has the echo.

Phase

Phase-triggers go off at specific points in the turns.

Examples: AEther Vial, see below.

There are 2 special parameters:

- Phase - The phase during which the trigger should go off.
- ValidPlayer - The player whose turn it should be.

There is 1 Triggered-variable:

- Player - The player whose turn it is.

PhaseIn & PhaseOut

Goes off when a permanent phases in or out, specifically only while that permanent is still on the battlefield since triggers don't work on phased out objects.

Examples: Shimmering Efreet, Ertai's Familiar.

There is 1 special parameter:

- ValidCard - The card phasing in or out must match this.

There is 1 Triggered-variable:

- Card - The card phasing in or out.

PlanarDice

Goes off when the planar dice is rolled in a Planechase game.

Examples: Panopticon, Orzhova.

There are 2 special parameters:

- ValidPlayer - The player that rolled the dice must match this.
- Result - The dice must roll this. Blank,Chaos or Planeswalk.

There is 1 Triggered-variable:

- Player - The player that rolled the dice.

PlaneswalkedTo & PlaneswalkedFrom

These triggers go off when a player planeswalks to or away from one or more planes.

Examples: Panopticon, Orzhova.

There is 1 special parameter:

- ValidCard - One of the planes must match this.

There is 1 Triggered-variable:

- Cards - A collection of all the cards planeswalked to or from.

Sacrificed

Sacrificed-triggers go off whenever a player sacrifices a permanent.

Examples: Dragon Appeasement, Mortician Beetle.

There are 2 special parameters:

- ValidPlayer - The player who sacrificed the card must match this.
- ValidCard - The sacrificed card must match this.

There is 1 Triggered-variable:

- Card - The card that was Sacrificed.

Scry

Goes off after a player has scryed and put the card(s) in the proper place.

Examples: Flamespeaker Adept, Knowledge and Power

There is 1 special parameter:

- ValidPlayer - The player that scryed must match this.

There is 1 Triggered-variable:

- Player - The player that scryed.

SearchedLibrary

Goes off when a player searches a library.

Examples: Ob Nixilis, Unshackled.

There are 2 special parameters:

- ValidPlayer - The player searching must match this.
- SearchOwnLibrary - If true, the player must be searching his or her own library (Didn't see that one coming, did ya? :P)

There is 1 Triggered-variable:

- Player - The player searching.

SetInMotion

Goes off when an Archenemy Scheme is set in motion.

Examples: A Display of My Dark Power, Perhaps You've Met My Cohort.

There is 1 special parameter:

- ValidCard - The scheme card that is set in motion must match this.

There is 1 Triggered-variable:

- Scheme - The scheme card that is set in motion.

Shuffled

Shuffled-triggers go off whenever a player shuffles his/her library.

Examples: Cofi's Trickster, Psychic Surgery.

There is 1 special parameter:

- ValidPlayer - The player whose turn it should be.

There is 1 Triggered-variable:

- Player - The player whose turn it is.

SpellCast, AbilityCast & SpellAbilityCast

Triggers go off whenever a spell, ability or either respectively is cast by either player.

Examples: AEther Barrier, Burning-Tree Shaman, Grip of Chaos.

There are 4 special parameters:

- ValidControllingPlayer - The player who controls the cast spell must match this.
- ValidActivatingPlayer - The player who activated the ability/spell must match this. (NOTE: For spells, the activator and controller are the same. They usually the same for abilities too, with the exception being abilities that can be activated by any player)
- ValidCard - The card the cast spell or ability originates from must match this.
- TargetsValid - If this parameter is present, the spell or ability must be targeted and at least one of it's targets must match this.

There are 4 Triggered-variable:

- Card - The card that the cast spell or ability originates from.
- SpellAbility - The SpellAbility object.
- Player - The player that controls the card that the cast spell or ability originates from.
- Activator - The player that activated the ability.

Taps & Untaps

These triggers go off when a permanent taps or untaps.

Examples: Artifact Possession, Frightshroud Courier.

There is 1 special parameter:

- ValidCard - The card that taps or untaps must match this.

There is 1 Triggered-variable:

- Card - The card that was tapped/untapped.

TapsForMana

This trigger goes off when a land is tapped for a mana ability.

Examples: Bubbling Muck, Market Festival.

There is 1 special parameter:

- ValidCard - The card that taps.

There are 3 Triggered-variable:

- Card - The card that was tapped.
- Player - the payer that did the tapping
- Produced - a String of the Mana produced by this tapping.

Transformed

Goes off when a card changes state from Original to Transformed or vice versa. (But not between any other 2 states)

Examples: Afflicted Deserter // Werewolf Ransacker, Huntmaster of the Fells // Ravager of the Fells.

There is 1 special parameter:

- ValidCard - The card that changes state must match this.

There is 1 Triggered-variable:

- Transformer - The card that changes state.

TurnFaceUp

Goes off when a card changes state from FaceDown to Original. (But not between any other 2 states)

Examples: Aphetto Exterminator, Fatal Mutation.

There is 1 special parameter:

- ValidCard - The card that changes state must match this.

There is 1 Triggered-variable:

- Card - The card that changes state.

Unequip

Goes off when an equipment card is unattached from a creature, wether it's voluntarily unattached or not.

Examples: Grafted Exoskeleton, Grafted Wargear.

There are 2 special parameters:

- ValidCard - The card that the equipment is unattached from must match this.
- ValidEquipment - The equipment being unattached must match this.

There are 2 Triggered-variables:

- Card - The card that the equipment is unattached from.
- Equipment - The equipment being unattached.

Vote

Goes off when a vote is called for, after all votes are cast.

Examples: Grudge Keeper.

There are no special parameters.

There is 1 Triggered-variable:

- OtherVoters - A collection of every player who voted, EXCEPT the controller of the trigger's host card.

Restrictions

You can use certain optional parameters to further restrict when a trigger should go off. These parameters are:

- TriggerZones - This parameter can be used to restrict which zone the card must be in in order for the trigger to go off. For example, if the trigger should only go off while the card is in the graveyard (As in Auntie's Snitch or Bridge from Below) you can use
`TriggerZones$ Graveyard`
- TriggerPhases - This parameter can be used to restrict the phases in which the trigger can trigger.
- OpponentTurn - This parameter can be used to restrict the trigger to only trigger on your opponents turn.(True/False)
- PlayerTurn - This parameter can be used to restrict the trigger to only trigger on your turn.(True/False)
- Metalcraft - If this parameter is set to "True", the controller of this card must have 3 or more artifacts on the battlefield for the trigger to go off.
- Threshold - As Metalcraft but requires the controller of the card to have 7 cards in the graveyard.
- PlayersPoisoned - This parameter specifies that a certain or both players must have at least 1 poison counter. Valid values are "You", "Opponent" or "Each".
- IsPresent - This parameter expects a [ValidCard](#) formula and only lets the trigger go off if there is a permanent on the battlefield that matches it.
- PresentCompare,PresentZone & PresentPlayer - These parameters only matter if the IsPresent parameter is used. They can be used to narrow down and how many valid cards must be present and where they must be.
- IsPresent2,PresentCompare2,PresentZone2 & PresentPlayer2 - Second requirement (see above).
- CheckSVar - Calculates the named SVar and compares the result against the accompanying SVarCompare parameter which takes the form of where is in LT,LE,EQ,NE,GE,GT.

Examples

Learn by example!

Contagion Clasp

(When Contagion Clasp enters the battlefield, put a -1/-1 counter on target creature.) The first thing to do is to identify the WHEN, i.e. when the trigger should go off so that we can decide on the proper mode. Here it should go off when a particular card moves from one zone into another, so `ChangesZone` is the logical mode here.

```
T:Mode$ ChangesZone
```

Next we look at which of `ChangesZone`'s 3 special parameters we want to make use of. Well, we want the trigger to go off when the card enters one specific zone, so we'll want to use the `Destination` parameter. Also, we only care about when it's a specific card being moved so we'll use `ValidCard`.

```
T:Mode$ ChangesZone | Destination$ Battlefield | ValidCard$
```

```
Card.Self
```

There, we've defined a trigger that goes off when this card moves from any zone (remember, `Origin` and `Destination` defaults to "Any") to the battlefield zone. But we still have to use two more parameters. First, "Execute". Execute should contain the name of the SVar that holds the Ability you want to be triggered. It can be any valid SVar name. I like to follow the convention "Trig" but your mileage may vary.

```
T:Mode$ ChangesZone | Destination$ Battlefield | ValidCard$
```

```
Card.Self | Execute$ TrigPutCounter
```

Lastly, "TriggerDescription". This is what will be put in the cards text box, i.e. the rules text. Always try to keep it as close to Oracle wording as possible. Also, you should use "CARDNAME" instead of the name of the card for convention.

```
T:Mode$ ChangesZone | Destination$ Battlefield | ValidCard$
```

```
Card.Self | Execute$ TrigPutCounter | TriggerDescription$ When
```

```
CARDNAME enters the battlefield, put a -1/-1 counter on target creature.
```

So we're all done, then? No, it's not a triggered **ability** until it actually has an ability. To define the ability we want to be triggered we simply use an [AbilityFactory](#) but instead of putting it on it's own line beginning with "A:", we put it in an SVar named what we put for the Execute parameter.

```
T:Mode$ ChangesZone | Destination$ Battlefield | ValidCard$
```

```
Card.Self | Execute$ TrigPutCounter | TriggerDescription$ When
```

```
CARDNAME enters the battlefield, put a -1/-1 counter on target creature.
```

```
SVar:TrigPutCounter:AB$PutCounter | Cost$ 0 | Tgt$ TgtC |
```

```
CounterType$ M1M1 | CounterNum$ 1
```

You may notice some strange things about the ability, namely the `Cost` parameter and the lack of a `SpellDescription` parameter. The reasoning for these things is that `AbilityFactory` requires non-drawback abilities to have a cost and drawback abilities not to have one. But if you want to do something like "When CARDNAME comes into play, you may pay 1 to...", that's where you'd use the `Cost` parameter here. The

SpellDescription is missing because triggers use their own
TriggerDescription instead.

AEther Vial

(At the beginning of your upkeep, you may put a charge counter on
AEther Vial.) Okay let's apply the procedure we learned when doing
Contagion Clasp:

Identify the WHEN - It should go off at the beginning of a specific
PHASE so, well, Phase mode it is!

```
T:Mode$ Phase
```

Next,look at Phase modes special parameters: Phase and ValidPlayer.
We can tell right away that Phase should be "Upkeep" and ValidPlayer
should be "You" since it should only trigger on your turn.

```
T:Mode$ Phase | Phase$ Upkeep | ValidPlayer$ You
```

Next up, we have to make sure that the trigger only goes off when
AEther Vial is in play. If it's not, why would we want to put counters on
it? We can use the TriggerZones parameter to restrict where the card
must be in order for the trigger to go off. It takes a comma-separated
list of zones but since we only want AEther Vial to trigger when it's on
the battlefield, we just give it "Battlefield".

```
T:Mode$ Phase | Phase$ Upkeep | ValidPlayer$ You | TriggerZones$  
Battlefield
```

Now then, there's one word in the rules text that changes things: "may".
To give the player a choice about wether or not to make use of the
triggered ability, we can use the OptionalDecider parameter. It's used
the same way as Defined parameters for AFs and the player it evaluates
to gets to decide if the trigger happens.

```
T:Mode$ Phase | Phase$ Upkeep | ValidPlayer$ You | TriggerZones$  
Battlefield | OptionalDecider$ You
```

Lastly as before, we add the Execute and TriggerDescription
parameters...

```
T:Mode$ Phase | Phase$ Upkeep | ValidPlayer$ You | TriggerZones$  
Battlefield | OptionalDecider$ You | Execute$ TrigPutCounter |  
TriggerDescription$ At the beginning of your upkeep, you may put a  
charge counter on CARDNAME.
```

...And our mad AbilityFactory-skills to write up the actual ability!

```
T:Mode$ Phase | Phase$ Upkeep | ValidPlayer$ You | TriggerZones$  
Battlefield | OptionalDecider$ You | Execute$ TrigPutCounter |  
TriggerDescription$ At the beginning of your upkeep, you may put a  
charge counter on CARDNAME.
```

```
SVar:TrigPutCounter:AB$PutCounter | Cost$ 0 | Defined$ Self |  
CounterType$ CHARGE | CounterNum$ 1
```

DING! Goblins are up!

Cinder Wall

(When Cinder Wall blocks, destroy it at end of combat.) This will be an example of a delayed trigger. A delayed trigger goes on the stack twice, once when the conditions are met, and once when the effect should happen. In Cinder Wall's case, once when it blocks, and once at the following end of combat step. To start with, we create a Blocks trigger that looks this.

```
T:Mode$ Blocks | ValidCard$ Card.Self
```

And this is where delayed triggers differ from ordinary triggers. We do NOT add an Execute parameter that points to an SVar containing the trigger's effect as an AF. Instead we add a DelayedTrigger parameter that points to an SVar containing ANOTHER trigger that specifies when the delayed effect should occur. This delayed trigger is built like an ordinary trigger and further points to an ability (Using the Execute parameter). Don't forget to omit the "T:" from the delayed trigger, for the same reason you omit "A:" from abilities in SVars!

```
T:Mode$ Blocks | ValidCard$ Card.Self DelayedTrigger$ DelTrig |
TriggerDescription$ When CARDNAME blocks, destroy it at end of
combat.
```

```
SVar:DelTrig:Mode$ Phase | Phase$ EndCombat | Execute$ TrigDestroy
| TriggerDescription$ Destroy CARDNAME.
```

```
SVar:TrigDestroy:AB$Destroy | Cost$ 0 | Defined$ Self
```

Test Cards

If you want some test cards for each trigger try these:

```
Always (Covetous Dragon)
DeclareAttack (Lightmine Field)
Attacks (Battle Cry)
Unblocked (Abyssal Nightstalker)
Blocked (Flanking)
Blocks (Wall of Junk)
Targeted (Tar Pit Warrior)
ChangeZones (Wall of Blossoms, AEther Flash, Valakut)
Phase (Bottomless Pit)
SpellAbilityCast (Dragonlair Spider)
Unequip (Grafted Wargear)
Sacrifice (Grave Pact)
Taps (Stonybrook Schoolmaster)
Untaps (Hollowsage)
Championed (Mistbind Clique/delayed due to input issue)
ChangeControllers (?)
CounterAdded (Flourishing Defenses)
CounterRemoved (Fungal Behemoth)
Clashed (Sylvan Echoes)
Cycled (Bant Sojourners)
Damaged (Akki Underminer, Living Artifact)
Discarded (Megrim)
Drawn (Underworld Dreams)
```



LandPlayed (Horn of Greed)
LifeGained (Ajani's Pridemate)
LifeLost (Mindcrank)
LoseGame
SetInMotion
Shuffled (Cosi's Trickster)
TapsForMana (Mana Flare)
Transformed (Afflicted Deserter)
TurnFaceUp (Bloodstoke Howler)

+ Add a custom footer

AbilityFactory

Katharina edited this page on Jul 20 · 4 revisions

EditNew page

[[TOC]]

NOTE: These factories are refactored from time to time (often to adapt to new sets), so while some entries could be slightly outdated, the base information should still be correct.

AbilityFactory parses differently from the Keyword parser. Your Ability line will look more like this:

```
A:{AB/SP/DB/ST}$ <AFSubclass> | {Necessary$ Parameters}|
{Separated$ By} | {Pipes$ Here}
```

In most cases, each AF subclass implements both the Spell and Ability. Much of the code is shared, so creating the data object will look very similar.

- AB is for Activated Abilities.
- SP is for Spell.
- DB is for Drawback and for many abilities that are subsidiary to other things, like replacements. They are only used to chain AFs together, and will never be the root AF.
- ST is for Static, this gets used in case the API should resolve without using the stack (e.g. special actions)

Common Parameters

Cost / UnlessCost

Cost\$ <AbilityCost> is the appropriate way to set the Cost of the Ability. Currently for spells, any additional costs including the original Mana cost need to appear in the Cost parameter in the AbilityFactory. For each card that uses it, the order in which the cost is paid will always be the same.

Secondary abilities such as the DB executed by triggers or replacements don't need costs. (This is one reason to use DB over AB in these cases.)

Read more about it in [Costs](#)

Target / Defined

Pages 52

Forge Wiki Home

- [User Guide](#)
 - [FAQ](#)
 - [Troubleshooting](#)
 - [AI](#)
 - [Network Play](#)
 - [Network FAQ](#)
 - [Network Extra](#)
- [Adventure Mode](#)
 - [Gameplay Guide](#)
 - [Getting Started](#)
 - [Towns & Capitals](#)
 - [Dungeons](#)
 - [Equipments and Items](#)
 - [Modding and Development:](#)
 - [Create Enemies](#)
 - [Create Rewards](#)
 - [Create Maps](#)
 - [Configure Planes](#)
- [Card Scripting API](#)
 - [Ability effects](#)
 - [Triggers](#)
 - [Replacements](#)
 - [Costs](#)
 - [Affected / Targets](#)
 - [Restrictions](#)
- [Development](#)
 - [IntelliJ Setup](#)
 - [Android Builds](#)
 - [Ownership](#)
 - [Simulation AI](#)
- [Customization and Themes](#)
 - Skins

There are two different ways to Target. One or the other will need to be used for Spells/Abilities that target. If your effect does not target, but instead defines what will be affected by the Spell/Ability, use the Defined Parameter.

Read more about it in [Affected / Targets](#)

Restriction

[Restriction](#) restricts when spells/abilities can be activated or resolved. Common examples are Putrid Leech's only activate this once per turn or different cards that can activate from Zones like the Hand or the Graveyard.

SpellDescription

SpellDescription is how the text of the ability will display on the card and in the option dialog for cards with multiple abilities.

The SpellDescription for secondary abilities (both AB and DB) is now displayed when (and if) the ability prompts for user input in the prompt pane so it is useful to put some nice text there.

StackDescription

(Optional) StackDescription is the description the ability will have on the stack. This is automatically generated by the effect, but may be overridden using this parameter. This is sometimes needed with complex effects, when the generated text can't handle some details. Properties of the spell can be accessed like this: {c:Targeted}. You can reuse the spell text by just putting `SpellDescription` or `None` to leave it empty.

AI instructions

IsCurse\$ True - For effects that are normally treated positive e. g. Pump

AITgts\$ BetterThanEvalRating.130

Factories

In Alphabetical Order.

AlterLife

AlterLife is for Abilities that Alter a player's life total.

GainLife

Have a player gain the specified amount of life.

- Music & Sounds

- [Card Images](#)
- [File Formats](#)

- [Missing Cards in Forge](#)
- [Credit and Thanks](#)

Clone this wiki locally

<https://github.com/Card-Forge>



A:AB\$ GainLife | Cost\$ T | LifeAmount\$ 1 | SpellDescription\$ You gain 1 life.

LifeAmount\$ is required. This is how much life you will gain.

Defined is optional, if it appears the defined player(s) gain life. Target is optional, if it appears and Defined doesn't then targeted player(s) gain life.

LoseLife

Have a player lose the specified amount of life.

A:AB\$ LoseLife | Cost\$ Sac<1/CARDNAME> | ValidTgts\$ Player | TgtPrompt\$ Target a player to lose a life | LifeAmount\$ 1 | SpellDescription\$ Target player loses 1 life.
A:SP\$ LoseLife | Cost\$ 2 B | Defined\$ Opponent | LifeAmount\$ 2 | SpellDescription\$ Each opponent loses 2 life.

LifeAmount\$ is required. This is how much life will be lost.

Target is optional. If Target doesn't appear then Defined will be used. Remember, if Defined is missing, the default for Players is "You"

Part of resolving sets the **SVar AFLifeLost** to the amount of life lost by all players.

Poison

Poison gives a player the specified number of poison counters.

A:AB\$ Poison | Cost\$ B T | ValidTgts\$ Player | TgtPrompt\$ Select target player | Num\$ 2 | SpellDescription\$ Target player gets 2 poison counters.

Parameters:

- Num (required) - the number of poison counters to give

Target is optional and used if available. Defined is optional and used if no Target (defaults to "You").

SetLife

SetLife sets one or both player's life total to a specified value (i.e. "your life total becomes 20" or "Target player's life total is equal to the number of cards in your graveyard").

A:SP\$ SetLife | Cost\$ 7 W W | ValidTgts\$ Player | TgtPrompt\$ Select target player | LifeAmount\$ 20 | SpellDescription\$ Target player's life total becomes 20.

Parameters:

LifeAmount (required) - the value to set the life total(s) to

Defined is optional. If it exists, it will be used. Target is optional. If it exists and defined doesn't it will be used. Default player is "You".

ExchangeLife

ExchangeLife switches the Life total of two players.

```
A:AB$ ExchangeLife | Cost$ 6 T | ValidTgts$ Player | TargetMin$ 2
| TargetMax$ 2 | TgtPrompt$ Select target player |
SpellDescription$ Two target players exchange life totals.
```

One of Defined or Target is required, since there needs to be two Players exchanging life If Defined it will be used. If Target exists and defined doesn't it will be used.

If there aren't two determined players by the SA, the activating player is added as the second player.

Animate

Animate handles animation effects like "This card becomes a 5/5 green creature with flying until end of turn." It is designed to handle color changing, type changing, P/T setting, and granting/removing abilities.

```
A:SP$Animate | Cost$ G | ValidTgts$ Land | TgtPrompt$ Select
target land | Power$ 3 | Toughness$ 3 | Types$ Creature |
SpellDescription$ Until end of turn, target land becomes a 3/3
creature that's still a land.
```

```
A:AB$Animate | Cost$ 1 B | Defined$ Self | Power$ 1 | Toughness$ 1
| Types$ Creature,Skeleton | Colors$ Black | Abilities$ ABRegen |
SpellDescription$ CARDNAME becomes a 1/1 black Skeleton creature
with "B: Regenerate this creature" until end of turn. It's still a
land. (If it regenerates, the next time it would be destroyed this
turn, it isn't. Instead tap it, remove all damage from it, and
remove it from combat.)
SVar:ABRegen:AB$Regenerate | Cost$ B | SpellDescription$
Regenerate CARDNAME.
```

```
A:AB$Animate | Cost$ 2 R G | Defined$ Self | Power$ 3 | Toughness$
3 | Types$ Creature,Elemental | Colors$ Red,Green | Triggers$
TrigAttack | SpellDescription$ Until end of turn, CARDNAME becomes
a 3/3 red and green Elemental creature with "Whenever this creature
attacks, put a +1/+1 counter on it." It's still a land.
SVar:TrigAttack:Mode$ Attacks | ValidCard$ Creature.Self |
Execute$ TrigPutCounter | TriggerDescription$ Whenever CARDNAME
attacks, put a +1/+1 counter on it.
SVar:TrigPutCounter:AB$PutCounter | Cost$ 0 | Defined$ Self |
CounterType$ P1P1 | CounterNum$ 1
```

Parameters:

- Power (required) - the power to assign to the animated card
- Toughness (required) - the toughness to assign to the animated card
- Types (optional) - the additional types to give the animated card; comma delimited
- OverwriteTypes (optional) - set to True if the animated being should have these types **instead** as opposed to **in addition to**
- RemoveTypes (optional) - a list of types to Remove from the animated card
- ChosenType (optional) - overrides types before it and just will add the ChosenType
- Keywords (optional) - a " & " delimited list of keywords to give the animated being (just like AB\$Pump)
- HiddenKeywords (optional) - a " & " delimited list of hidden keywords to give the animated being (just like AB\$Pump)
- RemoveKeywords (optional) - a " & " delimited list of keywords to remove from the animated being (just like AB\$Debuff)
- Colors (optional) - a comma-delimited list of Colors to give to the animated being (capitalized and spelled out) (ChosenColor accepted)
- Abilities (optional) - a comma-delimited list of SVar names which contain abilities that should be granted to the animated being
- OverwriteAbilities - Remove Abilities from animated being
- Triggers (optional) - a comma-delimited list of SVar names which contain triggers that should be granted to the animated being
- OverwriteTriggers - Remove/suppress triggers from animated being
- staticAbilities (optional) - a comma-delimited list of SVar names which contain static abilities that should be granted to the animated being
- OverwriteStatics- Remove static abilities from animated being
- OverwriteReplacements - Remove replacement effects from animated being
- RemoveAllAbilities - Remove all Abilities, Triggers, Statics, and Replacement effects
- sVars(optional) - a comma-delimited list of SVars that should be granted to the animated being
- Duration (Default is end of turn)
 - Permanent
 - UntilEndOfCombat - if the effect should last only until End of Combat instead of End of Turn
 - UntilHostLeavesPlay - if the effect should last as long as the host is still in play
 - UntilYourNextUpkeep
 - UntilControllerNextUntap

- UntilYourNextTurn

Target is optional, will be used if possible. Defined is optional, will be used if no Targets (Self by default)

Attach

Attach is being used directly only for Auras, primarily for Aura Spells, but also for Auras entering the battlefield by some effect.

AB\$ Attach | Cost\$ R R | ValidTgts\$ Creature | AILogic\$ Pump

Parameters:

- Object (optional) - This is the Object that will be Attached (generally the Source Card for Auras)
- AILogic - AI Logic tells the AI which base AI code it should use for Attaching
 - GainControl - Gains Control of the Attached Permanent (Control Magic)
 - Curse - A Generic Curse the AI has a handful of checks to see what the most appropriate Target is.
 - Pump - A Generic Pump. The AI has a handful of checks to see what the most appropriate Target is.
 - ChangeType - For Attachments that change types. Evil Presence is a good example. This logic should be expanded.
 - KeepTapped - For Attachments that keep a Permanent tapped. The AI will also check for a few things like Vigilance, and another KeepTapped Aura. Paralyzing Grasp is a good example.

Attach separates the actually granting of abilities from the attaching to permanents to streamline how things work.

BecomeMonarch

Bond

Soulbonding two creatures

Branch

Sometimes, an ability might do certain things when a specific condition is true, and other things if not. This can be implemented by using `Branch`. The branch evaluates the `SVar` specified by the property `BranchConditionSVar`, using the comparison defined with `BranchConditionSVarCompare` (such as `GT`, `LT`, etc). Depending on whether the condition evaluated to true or false, the subability defined by `TrueSubAbility` or `FalseSubAbility` is executed.

The example below is for "Composer of Spring", which allows either a "land" or a "land or creature" to be put on the battlefield, depending on the number of enchantments in play under your control.

```
SVar:TrigBranch:DB$ Branch | BranchConditionSVar$ X |  
BranchConditionSVarCompare$ GE6 | TrueSubAbility$  
PutLandCreature | FalseSubAbility$ PutLand  
SVar:PutLand:DB$ ChangeZone | Origin$ Hand | Destination$  
Battlefield | Tapped$ True | ChangeType$ Land.YouOwn  
SVar:PutLandCreature:DB$ ChangeZone | Origin$ Hand |  
Destination$ Battlefield | Tapped$ True | ChangeType$  
Creature.YouOwn, Land.YouOwn  
SVar:X:Count$Valid Enchantment.YouCtrl
```



ChangeState

Changing a cards State. This is mostly for Flip Cards or the Transform mechanic.

ChangeZone

ChangeZone is a united front of any card that changes zone. This does not include: drawing, discarding, destroying, or milling, as these represent specific words on which triggers and replacements can react. There are two primary forms, but the distinction is handled mostly in the codebase. The only thing that is required is to set appropriate parameters.

Origin and Destination are both required.

Origin is where the card is coming from.

Destination is where the card is going to. If Destination is Library, a LibraryPosition is recommended, but not required. **Default value of the LibraryPosition is 0.** 0 represents the top of the library, -1 represents the bottom.

There are two primary versions of ChangeZone.

Hidden Origin

The first is hidden, generally used for Origin zones that are not known information, like the Library or the Hand. The choice of "What card is changing zones?" happens during resolution.

A:SP\$ ChangeZone | Cost\$ W | Origin\$ Library | Destination\$ Library | LibraryPosition\$ 0 | ChangeType\$ Artifact,Enchantment | ChangeNum\$ 1 | SpellDescription\$ Search your library for an artifact or enchantment card and reveal that card. Shuffle your library, then put the card on top of it.

A:AB\$ ChangeZone | Cost\$ T | Origin\$ Hand | Destination\$ Battlefield | ChangeType\$ Land | ChangeNum\$ 1 | Optional\$ True | SpellDescription\$ You may put a land card from your hand onto the battlefield.

For Hidden, things like ChangeType and ChangeNum are used to restrict what can ChangeZone, and how many do. There are two parameters special to Hidden Origin:

"Chooser" defines which player has to decide which card changes zone (example You, Opponent).

"Mandatory" most of these abilities are not mandatory, but some are.

Known Origin

The second is known, generally used for Origin zones that are known information, like the Battlefield or the Graveyard. The choice of "What card is changing zones?" happens on activation, generally by targeting.

A:AB\$ ChangeZone | Cost\$ 1 U T | TgtPrompt\$ Choose target artifact card in your graveyard | ValidTgts\$ Artifact.YouCtrl | Origin\$ Graveyard | Destination\$ Library | SpellDescription\$ Put target artifact card from your graveyard on top of your library.

A:SP\$ ChangeZone | Cost\$ U U | ValidTgts\$ Permanent | TgtPrompt\$ Select target permanent | Origin\$ Battlefield | Destination\$ Hand | SpellDescription\$ Return target permanent to its owner's hand.

For Known, since it just uses Target, normal target parameters are used in this Scenario.

ChangeZoneResolve

This is a helper AF, for chained effects that create multiple permanents which should enter the battlefield at the same time.

To use it, you need to set the param "ChangeZoneTable" on the first effect and then call this at the end.

This is supported by the following effects:

- Amass
- CopyPermanent
- RepeatEach (*NOTE: if you wrap the creation, you don't need to call this AF on its own*)
- Token

Charm

This allows cards that have a mode to be chosen to occur after a trigger.

Parameters

- CharmNum - Number of Modes to Choose
- Choices - A Comma delimited list of SVars containing the Modes

Choose

ChooseType

This can be used when you are asked to choose a card type or creature type.

- Type - Required - Can be Card or Creature
- InvalidTypes - Optional - Use to specify any type that cannot be chosen (ex: "Choose any creature type except Wall")

The Defined is for target players.

NOTE_: Make sure that when this is used with a SubAbility that the DB\$AF_Whatever will support Card.ChosenType. This will need testing on a case by case basis.

Clash

This AF handles clashing. It takes two special parameters: WinSubAbility and OtherwiseSubAbility. They are both optional and work the same way, namely that it contains the name of an SVar that in turn contains a drawback to be executed. The example below is for Release the Ants.

```
A:SP$ DealDamage | Cost$ 1 R | Tgt$ TgtCP | NumDmg$ 1 |
SubAbility$ DBClash | SpellDescription$ Release the Ants deals 1
damage to target creature or player. Clash with an opponent. If you
win, return CARDNAME to its owner's hand.
SVar:DBClash:DB$ Clash | WinSubAbility$ DBReturn
SVar:DBReturn:DB$ ChangeZone | Defined$ Self | Origin$ Stack |
Destination$ Hand
```

Cleanup

A non-functional, maintenance AF used for Cleaning up certain Variables before a Spell finishes Resolving.

Parameters

- ClearRemembered\$ (optional) Set to True to clear this card's remembered list. Generally useful for Cards that Remember a card, do something to it, then need to forget it once it's done.

- ClearImprinted\$ (optional) Set to True to clear the list of imprinted cards.
- ClearChosenX\$ (optional) Set to True to clear the chosen X value.
- ClearTriggered\$ (optional) Set to True to clear any delayed triggers produced by this card.
- ClearCoinFlips\$ (optional) Set to True to clear the remembered coin flip result.
- ClearChosenCard\$ (optional) Set to True to clear the chosen cards.
- ForgetDefined\$ (optional) If present, remove the specified cards from this card's remembered list.

Copy

CopyPermanent

Copies a permanent on the battlefield.

Parameters:

- NumCopies - optional - the number of copies to put onto the battlefield. Supports SVar:X:????.
- Keywords - optional - a list of keywords to add to the copies
- AtEOT - optional
 - Sacrifice - set to this is copy should be sacrificed at End of Turn
 - Exile - set to this is copy should be exiled at End of Turn

CopySpellAbility

Copies a spell on the stack (Twincast, etc.).

Counter

Countering Spells or Abilities.

A:SP\$ Counter | Cost\$ 1 U | TargetType\$ Spell | TgtPrompt\$ Select target spell | ValidTgts\$ Card | UnlessCost\$ 3 | SpellDescription\$ Counter target spell unless its controller pays 3.

A:SP\$ Counter | Cost\$ U | TgtPrompt\$ Select target Activated or Triggered Ability | ValidTgts\$ Card | TargetType\$ Activated,Triggered | SpellDescription\$ Counter target activated or triggered ability.

A:SP\$ Counter | Cost\$ G | TargetType\$ Spell | ValidTgts\$ Instant,Aura | TargetValidTargeting\$ Permanent.YouCtrl | SpellDescription\$ Counter target instant or Aura spell that targets a permanent you control.

Parameters

- TargetType - Can be Spell,Activated,Triggered. If more than one, just put a comma in between.

- ValidTgts - a "valid" expression for types of source card (if you don't know what it is it's just "Card")
- TargetValidTargeting- a "valid" expression for targets of this spell's target
- Destination - send countered spell to: (only applies to Spells; ignored for Abilities)
 - Graveyard (Default)
 - Exile
 - TopDeck
 - Hand
 - BottomDeck
 - Shuffle

Counters

Factories to handle counters on cards.

PutCounter

Put any type of counter on a game object.

```
A:AB$ PutCounter | Cost$ T | CounterType$ CHARGE | CounterNum$1 |
SpellDescription$ Put a charge counter on CARDNAME.

A:SP$ PutCounter | Cost$ G | Tgt$ TgtC | CounterType$ P1P1 |
CounterNum$ 1 | SpellDescription$ Put a charge counter on CARDNAME.
```

Target is optional. If no Target is provided, the permanent will put counters on itself.

- CounterType (required) specifies the type of counter and should appear in all caps. It should be one of the values in the Counters enum.
- CounterNum (required) specifies how many counters will be put on the chosen card.

PutCounterAll

Put any type of counter on all valid cards.

- CounterType (required) specifies the type of counter and should appear in all caps. It should be one of the values in the Counters enum.
- CounterNum (required) specifies how many counters will be put on the chosen cards.
- ValidCards (required) specifies the cards to add counters to.

RemoveCounter

Remove any type of counter from a card.

Target is optional. If no Target is provided, the permanent will remove counters from itself.

- CounterType (required) specifies the type of counter and should appear in all caps. It should be one of the values in the Counters enum.
- CounterNum (required) specifies how many counters will be removed from the chosen card.
- UpTo is optional. If an effect states you may remove "up to X counters", set this to True.

RemoveCounterAll

Remove any type of counter from all valid cards.

- CounterType\$ (required) specifies the type of counter and should appear in all caps. It should be one of the values in the Counters enum.
- CounterNum\$ (required) specifies how many counters will be removed from the chosen cards.
- ValidCards\$ (required) specifies the card to remove counters from.

Proliferate

No own parameters.

MoveCounters

Used for cards that Move Counters on Resolution, requiring the Host card to have Counters for the Move to occur.

Parameters

- Source - The Source of the Moving Counters
- Defined - The Destination of the Moving Counters
- CounterType - The type of counter to move.
- CounterNum - The number of counters to move.

Damage

DealDamage

Deal damage to a specified player or permanent.

A:AB\$ DealDamage | Cost\$ T | Tgt\$ TgtCP | NumDmg\$ 1 |
SpellDescription\$ CARDNAME deals 1 damage to target creature or player.

NumDmg is required. This is the amount of damage dealt.

DamageAll

This is for damaging all applicable cards/players. Spells like Earthquake fit in this category.

```
A:SP$ DamageAll | Cost$ 1 B | NumDmg$ 1 | ValidCards$ Creature |  
ValidPlayers$ Each | ValidDescription$ each creature and each  
player. | SpellDescription$ CARDNAME deals 1 damage to each  
creature and each player.
```

ValidCards are for specifying the cards that will take damage.

ValidPlayers are for specifying the players.

Debuff

Parameters

- Keywords
- Duration

Debuff

An AbilityFactory for Removing Keywords, either temporarily or for longer durations.

DebuffAll

Same as Debuff, but for all of something

DelayedTrigger

Destroy

These APIs handles destruction of cards on the battlefield.

See [Sacrifice](#) for special case: 'Target opponent chooses [x cards].

Destroy those [cards]'

Destroy

```
A:SP$Destroy | Cost$ 1 W W | ValidTgts$ Artifact,Enchantment |  
TgtPrompt$ Select target artifact or enchantment |  
SpellDescription$ Destroy target artifact or enchantment.
```

DestroyAll

```
A:SP$ DestroyAll | Cost$ 3 U | ValidCards$ Forest |  
SpellDescription$ Destroy all Forests.
```

Effect

Effect is an oddball of the AF family. Where usually AFs have similarities to each other to help with AI use, Effect doesn't fall under that jurisdiction. In general, an effect is some type of SA that lasts longer than its resolution.

A good example is High Tide. For the rest of the turn, High Tide makes all Islands produce an extra mana. It doesn't matter if the Island was in play, if it turned into an Island after High Tide was cast, any of that.

```
A:SP$ Effect | Cost$ U | Name$ High Tide Effect | Triggers$
IslandTrigger | SVars$ TrigMana | SpellDescription$ Until end of
turn, whenever a player taps an Island for mana, that player adds U
to his or her mana pool (in addition to the mana the land
produces).
```

```
SVar:IslandTrigger:Mode$ TapsForMana | ValidCard$ Island |
Execute$ TrigMana | TriggerDescription$ Whenever a player taps an
Island for mana, that player adds U to his or her mana pool (in
addition to the mana the land produces).
```

```
SVar:TrigMana:AB$Mana | Cost$ 0 | Produced$ U | Amount$ 1
```

Effect is most similar to Token as it creates a pseudo-permanent, except Effect creates in the command zone rather than the battlefield. It stays active there for a set Duration.

Parameters

- Abilities,Triggers,SVars are comma separated lists which contain SVars that point to the appropriate type that the Effect will gain.
- Duration is how long the Effect lasts. Right now, most effects will last Until End of Turn. In the future, they may have other conditions.

Duration\$ Permanent for effects that have no specific Duration.

- Stackable\$ False - Most Effects are assumed to be Stackable. By setting the Stackable Flag to False, the AI will know having a second one in play is useless, so will save it's Resource for something else.
- Image - a file_name_without_extension (image needs to reside in the tokens directory)

Game outcome

GameDraw

GameLoss

GameWin

RestartGame

Used in the script of Karn Liberated

Explore

Fight

Fog

Fog is an ability based on the original Fog spell. "Prevent all combat damage that would be dealt this turn." While this could be done with an effect, the specialized nature of the AI gives it it's own AF.

GainControl

Example: Act of Aggression

```
A:SP$ GainControl | Cost$ 3 PR PR | ValidTgts$ Creature.OppCtrl |
TgtPrompt$ Select target creature an opponent controls. |
LoseControl$ EOT | Untap$ True | AddKWs$ Haste | SpellDescription$
Gain control of target creature an opponent controls until end of
turn. Untap that creature. It gains haste until end of turn.
```

Parameters:

- NewController(Targeted player, if there is no target player, the default is the ActivatingPlayer)
- AllValid(all valid types, no targets)
- LoseControl(LeavesPlay, Untap, LoseControl, EOT(end of turn))

Goad

Investigate

Mana

For lands or other permanent to produce mana.

```
A:AB$ Mana | Cost$ T | Produced$ <ManaType> | SpellDescription$
Add W to your mana pool.
```

In this example ManaType would be W.

Manifest

PermanentState

API for things that alter a permanent's state: tap, untap, or phase in/out.

Untap

A:AB\$ Untap | Cost\$ G | ActivationLimit\$ 1| SpellDescription\$ Untap CARDNAME. Activate this ability only once each turn.

A:SP\$ Untap | Cost\$ W | ValidTgts\$ Permanent | TgtPrompt\$ Select target permanent | SpellDescription\$ Untap target permanent.

Target is optional. If not provided, will untap the permanent with this ability.

UntapAll

A:SP\$ UntapAll | Cost\$ G | ValidCards\$ Creature.YouCtrl | SpellDescription\$ Untap all creatures you control.

Tap

A:AB\$ Tap | Cost\$ R | ValidTgts\$ Wall | TgtPrompt\$ Select target wall | SpellDescription\$ Tap target wall.

TapAll

A:AB\$ TapAll | Cost\$ U U U | ValidCards\$ Creature.withoutFlying | SpellDescription\$ Tap all creatures without flying.

TapOrUntap

Phases

Play

Playing a card or cards as part of SA. The player may have to make a choice about which card to play if there are more choices than the number of cards to play.

Sunbird's Invocation uses many of the parameters.

Amount - How many cards can be played (a non-negative integer or "All").

Valid - Selection criteria for valid cards from the zone to cast.

ValidSA - Applied after Valid, this will filter based on all spells of the cards.

ValidZone - The zone to look in to determine the valid cards.

ShowCards - Other cards in the zone to show when selecting valid cards.

Optional - Casting the card is optional.

RememberPlayed - Remember the card played.

ForgetRemembered - Remove all remembered cards from the source (but only if a card was successfully played)

ForgetTargetRemembered - Remove the played card from remembered cards (but only if it was successfully played)

WithoutManaCost - The card can be cast without mana payment.

PreventDamage

AF damage prevention effects.

PreventDamage

```
- A:SP$ PreventDamage | Cost$ W | ValidTgts$ Creature | Amount$ 3 | TgtPrompt$ Select target creature | SpellDescription$ Prevent the next 3 damage that would be dealt to target creature this turn.
```



PreventDamageAll

Same as PreventDamage, but for all ValidCards and/or ValidPlayers

Protection

Protection

Protection grants protection from colors or cards types, or creature types. Anything that has "Protection from ..."

Gains - required - the thing to gain protection from (green, artifacts, Demons, etc.) or "Choice" if you can choose one of...

Choices - optional

- if Gains\$ Choice then this is a comma-delimited list of choices

ProtectionAll

Same as Protection, but for all ValidCards and/or ValidPlayers

Pump

Pump

This factory handles pumping creatures power/toughness or granting abilities to permanents (usually creatures).

```
- A:AB$ Pump | Cost$ R | NumAtt$ +1 | SpellDescription$ CARDNAME gets +1/+0 until end of turn.
- A:SP$ Pump | Cost$ 1 U | ValidTgts$ Creature | KW$ Shroud|
```



SpellDescription\$ Target creature gains shroud until end of turn. | TgtPrompt\$ Select target creature.

Target is optional. If it's not provided, the activating permanent will be pumped.

NumAtt\$ is optional, will pump Power.

NumDef\$ is optional, will pump Toughness.

KW\$ is optional, will give temporary keywords.

PumpAll

Regenerate

Regenerate

Regenerate is similar to abilities like Pump. But for creating Regeneration shields.

```
- A:AB$ Regenerate | Cost$ B | SpellDescription$ Regenerate  
CARDNAME  
- A:SP$ Regenerate | Cost$ W | ValidTgts$ Creature |  
TgtPrompt$  
Select target creature | SpellDescription$ Regenerate  
target  
creature.
```



Target is optional. If not provided, will regenerate the permanent with this ability.

RegenerateAll

Same as regenerate, but for all.

ValidCards - required - a valid expression for cards to regenerate

Repeat

Repeat the specified ability.

Repeat

```
A:SP$ Repeat | Cost$ 3 B B | RepeatSubAbility$ DBDig |  
RepeatOptional$ True
```

- MaxRepeat - optional - the maxium times to repeat, execute repeat ability at least once
- RepeatSubAbility - required - setup subability to repeat
- RepeatOptional - optional - you make the choice whether to repeat the process

- RepeatPresent, RepeatCompare, RepeatDefined, RepeatCheckSVar, RepeatSVarCompare - optional - condition check

RepeatEach

- RepeatSubAbility - required - to set up repeat subability
- RepeatCards - to repeat for each valid card (zone: present zone of the valid repeat cards, default: battlefield)
- DefinedCards
- RepeatPlayers - to repeat for each valid player
- RepeatCounters - to repeat for each valid counters

Reveal

Dig

Dig is for an ability that does basically this: "You look at the X cards of your Library, put Y of them somewhere, then put the rest somewhere." Think of Impulse.

- DigNum - Required - look at the top number of cards of your library.
- Reveal - Optional - for abilities that say "Reveal the top X cards of your library". Default is false.
- SourceZone - Optional - the zone to dig in. Default is Library.
- DestinationZone - Optional - the zone to put the Y cards in. Default is Hand.
- LibraryPosition - Optional - if DestinationZone is Library, use this to specify position. Default is -1 (bottom of library).
- ChangeNum - Optional - the number of cards to move to the DestinationZone (or "All" when it's for things like "put all lands revealed this way into your hand"). Default is 1.
- ChangeValid - Optional - use this to specify if "you may move an artifact to DestinationZone". Default is any Card.
- AnyNumber - Optional - use if you can move any number of Cards to DestinationZone. Default is false. (think of Lead the Stampede)
- Optional - Optional - set this if you "may" move a card to DestinationZone. Default is false.
- DestinationZone2 - Optional - the zone to put the rest of the cards in. If it is library, you are prompted for the order. Default is Library.
- LibraryPosition2 - Optional - if DestinationZone2 is Library, use this to specify position. Default is -1 (bottom of library).

DigUntil

RevealHand

Look at a player's hand.

Target or Defined is required.

```
A:AB$ RevealHand | Cost$ T | ValidTgts$ Player | TgtPrompt$ Select  
target player | SpellDescription$ Look at target player's hand.
```

Scry

```
A:AB$ Scry | Cost$ 1 T | ScryNum$ 2
```

RearrangeTopOfLibrary

Reveal

```
A:AB$ Reveal | Cost$ 2 U T | Defined$ You | RevealValid$ Card.Blue  
| AnyNumber$ True | RememberRevealed$ True
```

Parameters:

- RevealValid: to limit the valid cards.
- AnyNumber
- Random
- RememberRevealed: to remember the cards revealed

PeekAndReveal

This AF is very similar to things that Dig can do, but handle a much simpler form, with less complex coding underneath. Similar to how RearrangeTopOfLibrary could be handled with Dig.

Primarily used with cards that allow you to Peek at the top card of your library, and allow you to reveal it if it's of a certain type. The Kinship cards fit this bill perfectly, so they are used to simplify the complex popups that would be required if using multiple Dig SubAbilities.

RevealOptional - Whether or not the Reveal is optional.

RememberRevealed - Whether to remember the revealed cards (after filtering by Valid)

RememberPeeked - Whether to remember the peeked cards (only if they are not revealed!)

RevealValid - defaults to Card, but allows you to set a specific ValidType if you can only have certain things

PeekAmount - defaults to 1, but allows you to peek at multiple cards if possible

RollDice

Sacrifice

Sacrifice

Usually you choose a player and that player has to sacrifice something

```
A:SP$ Sacrifice | Cost$ 1 B | ValidTgts$ Player | SacValid$  
Creature | SacMessage$ Creature | Amount$ 2 | SpellDescription$  
Target player sacrifices a creature.
```

Destroy\$ True - An optional parameter for destroying permanents
target player chooses (eg: Burning of Xinye, or Imperial Edict).

```
A:SP$ Sacrifice | Cost$ 1 B | ValidTgts$ Opponent | SacValid$  
Creature | SacMessage$ Creature | Destroy$ True | SpellDescription$  
Target opponent chooses a creature he or she controls. Destroy it.
```

SacrificeAll

StoreSVar

Token

Token simply lets you create tokens of any type.

```
A:SP$ Token | Cost$ 3 W U | TokenImage$ W 1 1 Bird Flying |  
TokenAmount$ X | TokenName$ Bird | TokenTypes$ Creature,Bird |  
TokenOwner$ You | TokenColors$ Blue | TokenPower$ 1 |  
TokenToughness$ 1 | TokenKeywords$ Flying
```

This ability factory does not take a target. All the parameters are mandatory except for TokenKeywords. If you provide a non-integer for TokenAmount, TokenPower or TokenToughness the AF will attempt to look for an SVar of that name and interpret it's contents as a Count\$ line. Worth noting is that TokenTypes and TokenColors are simple commaseparated lists while TokenKeywords is a list where the items are separated by "<>". If TokenImage is not provided, the factory will attempt to construct a filename on it's own. TokenOwner can use Defined-like parameters, such as "You" "Opponent" or the new Triggered-Variables.

You can also use the parameters TokenAbilities\$, TokenTriggers\$ and TokenSVars\$ to give the created tokens any number of either. For example, here's how Growth Spasm creates an Eldrazi Spawn token complete with ability.

```
SVar:DBToken:DB$Token | TokenAmount$ 1 | TokenName$ Eldrazi  
Spawn | TokenTypes$ Creature,Eldrazi,Spawn | TokenOwner$ You |  
TokenColors$ Colorless | TokenPower$ 0 | TokenToughness$ 1 |  
TokenAbilities$ ABMana SVar:ABMana:AB$Mana | Cost$  
Sac<1/CARDNAME> | Produced$ 1 | Amount$ 1 | SpellDescription$  
Add 1 to your mana pool.
```



As another example, here's Mitotic Slimes' use of TokenTriggers\$:

T:Mode\$ ChangesZone | Origin\$ Battlefield | Destination\$ Graveyard | ValidCard\$ Card.Self | Execute\$ TrigTokenSenior | TriggerDescription\$ When CARDNAME is put into a graveyard from the battlefield, put two 2/2 green Ooze creature tokens onto the battlefield. They have "When this creature is put into a graveyard, put two 1/1 green Ooze creature tokens onto the battlefield."

SVar:TrigTokenSenior:AB\$ Token | Cost\$ 0 | TokenImage\$ g 2 2 ooze | TokenName\$ Ooze | TokenTypes\$ Creature,Ooze | TokenColors\$ Green | TokenOwner\$ You | TokenPower\$ 2 | TokenToughness\$ 2 | TokenAmount\$ 2 | TokenTriggers\$ TriggerJunior | TokenSVars\$ TrigTokenJunior
SVar:TriggerJunior:Mode\$ ChangesZone | Origin\$ Battlefield | Destination\$ Graveyard | ValidCard\$ Card.Self | Execute\$ TrigTokenJunior | TriggerDescription\$ When this creature is put into a graveyard, put two 1/1 green Ooze creature tokens onto the battlefield. SVar:TrigTokenJunior:AB\$Token | Cost\$ 0 | TokenImage\$ g 1 1 ooze | TokenName\$ Ooze | TokenTypes\$ Creature,Ooze | TokenColors\$ Green | TokenOwner\$ You | TokenPower\$ 1 | TokenToughness\$ 1 | TokenAmount\$ 2

Turn structure

AddPhase

AddTurn

A:SP\$ AddTurn | Cost\$ 1 U | NumTurns\$ 1 | SpellDescription\$ Take an extra turn after this one.

EndTurn

ReverseTurnOrder

SkipPhase

SkipTurn

ZoneAffecting

For specific effects that handle zones in a specific manner

Draw

A:AB\$ Draw | Cost\$ 1 Sac<1/CARDNAME> | NumCards\$ 1 | SpellDescription\$ Draw a card.

Discard

A:AB\$ Discard | Cost\$ T | ValidTgts\$ Opponent | NumCards\$ 1 | Mode\$ TgtChoose | SpellDescription\$ Target opponent discards a card.

- NumCards - the number of cards to be discarded (may be integer or X)
- Mode - the mode of discard - should match spDiscard
 - Random
 - TgtChoose
 - RevealYouChoose
 - Hand
- DiscardValid - a ValidCards syntax for acceptable cards to discard
- UnlessType - a ValidCards expression for "discard X unless you discard "

Mill

A:AB\$ Mill | Cost\$ 2 T | NumCards\$ 2 | ValidTgts\$ Player | TgtPrompt\$ Choose a player to mill | SpellDescription\$ Target player puts the top two cards of his or her library into his or her graveyard.

Shuffle

Used for shuffling a player's library

- Optional - Set this parameter if the user should be prompted Yes/No to shuffle the given library. Default is false.

TwoPiles

Vote

+ Add a custom footer

Card scripting API

Chris H edited this page on Jul 7 · 8 revisions

EditNew page

A reference guide for Scripting Cards using the API parsed by the Forge Engine.

Base Structure 🔗

By opening any file in the /res/cardsfolder folder you can see the basic structure of how the data is created. Here's an example of a vanilla creature:

```
Name:Vanilla Creature
ManaCost:2 G
Types:Creature Beast
Text:no text
PT:2/2
```




The name of this card is Vanilla Creature. It's casting cost is 2G. It has the types Creature and Beast. It will not display any additional text in the card's template. It has a Power-Toughness of 2/2. The text line appears only if any rules printed on card are not backed by abilities defined for the given card (in most cases presence of Text line means such abilities are hardcoded).

If a card has two faces, use AlternateMode:{CardStateName} in the front face and separate both by a new line with the text "ALTERNATE".

There are a few other Parameters that will appear in most, if not all, cards. These are

Property	Description
A	Ability effect
AI	<div>RemoveDeck:</div> <div>* All</div> <div>This will prevent the card from appearing in random AI decks. It is applicable for cards the AI can't use at all like Dark Ritual and also for cards that the AI could use, but only ineffectively like Tortoise Formation. The AI won't draft these cards.</div> <div>* Random</div> <div>This will prevent the card from appearing in random</div>

▶ Pages 52

Forge Wiki Home 

User Guide

- [FAQ](#)
- [Troubleshooting](#)
- [AI](#)
- [Network Play
 - \[Network FAQ\]\(#\)
 - \[Network Extra\]\(#\)](#)

Adventure Mode

- [Gameplay Guide
 - \[Getting Started\]\(#\)
 - \[Towns & Capitals\]\(#\)
 - \[Dungeons\]\(#\)
 - \[Equipments and Items\]\(#\)](#)
- [Modding and Development:
 - \[Create Enemies\]\(#\)
 - \[Create Rewards\]\(#\)
 - \[Create Maps\]\(#\)
 - \[Configure Planes\]\(#\)](#)

Card Scripting API

- [Ability effects](#)
- [Triggers](#)
- [Replacements](#)
- [Costs](#)
- [Affected / Targets](#)
- [Restrictions](#)

Development

- [IntelliJ Setup](#)
- [Android Builds](#)
- [Ownership](#)
- [Simulation AI](#)

Customization and Themes


- [Skins](#)

Property	Description
	<p>decks. It is only applicable for cards that are too narrow for random decks like Root Cage or Into the North. The AI won't draft these cards.</p> <p>* <code>NonCommander</code></p>
Colors	<p>Color(s) of the card</p> <p>When a card's color is determined by a color indicator rather than shards in a mana cost, this property must be defined. If no identifier is needed, this property should be omitted.</p> <p>* <code>Colors:red</code> - This is used on Kobolds of Kher Keep, which has a casting cost of {0} and requires a red indicator to make it red.</p> <p>* <code>Colors:red,green</code> - Since Arlinn, Embraced by the Moon has no casting cost (it's the back of a double-faced card), the red and green indicator must be included.</p>
DeckHints	<p>AI-related hints for a deck including this card</p> <p>To improve synergy this will increase the rank of of all other cards that share some of its DeckHints types. This helps with smoothing the selection so cards without these Entries won't be at an unfair disadvantage.</p> <p>The relevant code can be found in the CardRanker class.</p>
DeckNeeds	<p>This can be considered a stronger variant when the AI should not put this card into its deck unless it has whatever other type is specified. The way this works is "inverted": it will directly decrease the rank of the card unless other cards are able to satisfy its types. If a card demands more than one kind of type you can reuse it:</p> <p><code>DeckNeeds:Type\$Human & Type\$Warrior</code> will only find Human Warrior compared to</p> <p><code>DeckNeeds:Type\$Human Warrior</code> which is either</p>
DeckHas	<p>specifies that the deck now has a certain ability (like, token generation or counters) so that the drafting/deckbuilding AI knows that it now meets requirements for DeckHints/DeckNeeds. This is actually very useful since many of these (such as <code>Ability\$Graveyard</code>, <code>Ability\$Token</code>, <code>Ability\$Counters</code>)</p>

- [Music & Sounds](#)
- [Card Images](#)
- [File Formats](#)

- [Missing Cards in Forge](#)
- [Credit and Thanks](#)

Clone this wiki locally

<https://github.com/Card-Forge>


Property	Description
	<p>are not deduced by parsing the abilities, so an explicit hint is necessary. Using the other types is also supported in case the implicit parsing wouldn't find it.</p> <p>It doesn't require exact matching to have an effect but cards that care about multiple entries for a given type will be judged higher if a card seems to provide even "more" synergy for it.</p> <p>Example: Chishiro has two abilities so <code>DeckHas:Ability\$Token & Ability\$Counters</code> is used, therefore score for <code>DeckNeeds:Ability\$Token Counters</code> is increased</p>
K	Keyword
Loyalty	Number of starting loyalty counters
ManaCost	<p>Cost to cast the card shown in mana shards</p> <p>This property is required. It has a single parameter that is a mana cost.</p> <p>* <code>ManaCost:no cost</code> for cards that cannot be cast</p> <p>* <code>ManaCost:1 W W</code> sets the casting cost to $\{1\}\{W\}\{W\}$</p>
Name	<p>Name of the card</p> <p>A string of text that serves as the name of the card. Note that the registered trademark symbol cannot be included, and this property must have at least one character.</p> <p>Example: * <code>Name:A Display of My Dark Power</code> sets the card's name to "A Display of My Dark Power"</p>
Oracle	<p>The current Oracle text used by the card.</p> <p>We actually have a Python Script that runs to be able to fill in this information, so don't worry about manually editing a lot of cards when Wizards decides to change the rules.</p> <p>This field is used by the Deck Editor to allow non-Legendary Creatures to be marked as potential commanders. Make sure "CARDNAME can be your commander." appears in the oracle text.</p>
PT	Power and toughness

Property	Description
R	Replacement effect
S	Static ability
SVar	String variable. Used throughout scripting in a handful of different ways.
T	Triggered ability
Text	Additional text that needs to be displayed on the CardDetailPanel that doesn't have any spell/ability that generates a description for it, for example "CARDNAME can be your commander." or "X can't be 0."
Types	Card types and subtypes Include all card types and subtypes, separated by spaces. Example: * Types:Enchantment Artifact Creature Golem for a card that reads Enchantment Artifact Creature -- Golem

Rarity and Set info are now defined in edition definition files. These can be found at /res/reditions path.

General SVars [↗](#)

SVar:SoundEffect:goblinpolkaband.mp3

The sound system supports a special SVar that defines the sound that should be played when the spell is cast.

SVar:AltCost:[cost]

This SVar is for cards that have an Alternate cost, such as Force of Will. You are allowed to pay the Alternate Cost instead of the normal Mana cost when casting this spell.

SVar:AIPreference:SacCost\$Creature.token,Creature.cmcLE2

SVar:AntiBuffedBy:[ValidCards]

If a permanent with this SVar is on the battlefield under human control the AI will play the specified cards in Main1. Applicable for cards like Heart Sliver or Timid Drake.

SVar:BuffedBy:[ValidCards]

If a permanent with this SVar is on the battlefield under its control the AI will play the specified cards in Main1. Applicable for creatures with a P/T setting static ability (Kithkin Rabble) or additional buffes (Radiant, Archangel).

SVar:EnchantMe:[Multiple]/[Once]

Creatures with "Multiple" in this SVar will always be preferred when the AI enchants (Rabid Wombat), creatures with "Once" only if they are not enchanted already (Gate Hound).

SVar:EquipMe:[Multiple]/[Once]

Creatures with "Multiple" in this SVar will always be preferred when the AI equippes (Myr Adapter), creatures with "Once" only if they are not equipped already (Kor Duelist).

SVar:AIEvaluationModifier:[ValidAmount]

SVar:EndOfTurnLeavePlay:True

SVar:maxLevel:

SVar:HasCombatEffect:True

SVar:HasAttackEffect:True

SVar:HasBlockEffect:True

SVar:MustAttack:True

SVar:MustBeBlocked:True

SVar:NeedsToPlayVar:[ValidCards]

SVar:ManaNeededToAvoidNegativeEffect:

SVar:NonStackingEffect:True

SVar:PlayMain1:TRUE/ALWAYS/OPPONENTCREATURES

The AI will play cards with this SVar in its first main phase. Without other AILogic, it will usually not play any permanents without this in Main1.

SVar:SacMe:[number]

The AI will sacrifice these cards to pay costs. The higher the number the higher the priority. Example: Hatching Plans has SVar:SacMe:5.

SVar:Targeting:Dies

SVar:UntapMe:True

The AI will prioritize untapping of this card.

SVar:AIUntapPreference:

Count is our general value computation function. It's quite varied with a lot of different things it can calculate and is often being updated.

Keywords [↗](#)

All keywords need to be prepended with "K:" to be parsed correctly. Each keyword must appear on a separate line.

Keywords without Parameters [↗](#)

This section is for Keywords that require no additional parameters and are one or two words long. Most of these you would see exactly on cards in the game.

- Cascade
- Changeling
- Cipher
- Conspire
- Convoke
- Deathtouch
- Defender
- Delve
- Devoid
- Double Strike
- Epic
- Exalted
- Fear
- First Strike
- Flanking
- Flash
- Flying
- Forestwalk
- Fuse
- Haste
- Hideaway
- Horsemanship
- Indestructible
- Infect
- Intimidate
- Islandwalk
- Landfall
- Legendary landwalk

- Lifelink
- Living Weapon
- Menace
- Mentor
- Mountainwalk
- Nonbasic landwalk
- Persist
- Plainswalk
- Prowess
- Provoke
- Reach
- Rebound
- Retrace
- Riot
- Shadow
- Shroud
- Snow forestwalk
- Snow islandwalk
- Snow landwalk
- Snow mountainwalk
- Snow plainswalk
- Snow swamp walk
- Soulbond
- Split second
- Storm
- Sunburst
- Swampwalk
- Totem Armor
- Trample
- Unblockable
- Undying
- Vigilance
- Wither

Keywords with parameters [↗](#)

- Adapt:{cost}
- AdjustLandPlays:{params}
- Afterlife:{N}
- AlternateAdditionalCost:{cost}
- Amplify:{cost}:{validType(comma separated)}
- Annihilator:{magnitude}

- Bloodthirst:{magnitude}
- Bestow:{cost}
- Bushido:{magnitude}
- CantBeBlockedByAmount {xMath}
- Champion:{validType}
- CostChange:{params}
- Crew:{cost}
- Cumulative upkeep:{cost}:{Description}
- Cycling:{cost}
- Dash:{cost}
- Devour:{magnitude}
- Dredge:{magnitude}
- Echo:{cost}
- Emerge:{cost}
- Enchant {params} [Curse]
- Enchant {Type}
- Entwine:{cost}
- Equip:{cost}
- etbCounter:{CounterType}:{CounterAmount}
- ETBReplacement:{Control/Copy/Other}:{AbilitySVar}[:Optional]
- Evoke:{cost}
- Fabricate:{cost}
- Fading:{FadeCounters}
- Flashback:{cost}
- Foretell:{cost}
- Fortify:{cost}
- Graft:{value}
- Haunt:{ability}:{Description}
- Hexproof:{ValidCards}:{Description}
- Kicker:{cost}
- Level up:{cost}
- Madness:{cost}
- ManaConvert:
- maxLevel:{magnitude}
- MayEffectFromOpeningHand:{Effect}
- Miracle:{cost}
- Modular:{magnitude}
- Monstrosity:{cost}
- [Mega]Morph:{cost}
- Multikicker:{magnitude}
- Mutate:{cost}
- Ninjutsu:{cost}

- Outlast:{cost}
- Partner:{CardName}
- Poisonous {magnitude}
- PreventAllDamageBy {ValidCards}
- Protection:{ValidCards}:{Description}
- Prowl:{cost}
- Rampage:{magnitude}
- Recover:{cost}
- Renown:{N}
- Replicate:{cost}
- Ripple:{magnitude}
- Soulshift:{magnitude}
- Strive:{cost}
- Suspend:{turns}:{cost}
- Transmute:{cost}
- Toxic:{poisonCounters}
- TypeCycling:{Type}:{cost}
- Unearth:{cost}
- UpkeepCost:{cost}
- Vanishing:{TimeCounters}

Longer Card Properties [↗](#)

Plaintext properties/triggers. This section is for Keywords that are two words or longer. CARDNAME is replaced by the card's name.

- All creatures able to block CARDNAME do so.
- CARDNAME assigns no combat damage
- CARDNAME attacks each turn if able.
- CARDNAME attacks each combat if able.
- CARDNAME blocks each combat if able.
- CARDNAME blocks each turn if able.
- CARDNAME can attack as though it didn't have defender.
- CARDNAME can attack as though it had haste.
- CARDNAME can block as though it were untapped.
- CARDNAME can block creatures with shadow as though they didn't have shadow.
- CARDNAME can block creatures with landwalk abilities as though they didn't have those abilities.
- CARDNAME can block only creatures with flying.
- CARDNAME can only attack alone.
- CARDNAME can't attack.
- CARDNAME can't attack if you cast a spell this turn.

- CARDNAME can't attack if defending player controls an untapped creature with power {rest of text string}
- CARDNAME can't attack or block.
- CARDNAME can't attack or block alone.
- CARDNAME can't be countered.
- CARDNAME can't be enchanted.
- CARDNAME can't be equipped.
- CARDNAME can't be regenerated.
- CARDNAME can't be the target of Aura spells.
- CARDNAME can't block.
- CARDNAME can't block creatures with power {rest of text string}
- CARDNAME can't block unless a creature with greater power also blocks.
- CARDNAME can't block unless at least two other creatures block.
- CARDNAME can't transform
- CARDNAME doesn't untap during your untap step.
- CARDNAME enters the battlefield tapped.
- CARDNAME is {color}.
- CARDNAME must be blocked if able.
- CARDNAME must be blocked by exactly one creature if able.
- CARDNAME must be blocked by two or more creatures if able.
- CARDNAME can't be blocked unless all creatures defending player controls block it.
- CARDNAME's power and toughness are switched
- CARDNAME untaps during each other player's untap step.
- CARDNAME's activated abilities can't be activated.
- Creatures with power greater than CARDNAME's power can't block it.
- Creatures can't attack unless their controller pays:{params}
- Damage that would be dealt by CARDNAME can't be prevented.
- Damage that would reduce your life total to less than 1 reduces it to 1 instead.
- Enchant artifact
- Enchant creature
- Enchant creature with converted mana cost 2 or less
- Enchant creature without flying
- Enchant red or green creature
- Enchant land
- Enchant land you control
- Enchant tapped creature
- No more than one creature can attack each combat.
- No more than one creature can block each combat.

- No more than two creatures can attack you each combat.
- No more than two creatures can block each combat.
- Play with your hand revealed.
- Prevent all combat damage that would be dealt to and dealt by CARDNAME.
- Prevent all combat damage that would be dealt by CARDNAME.
- Prevent all combat damage that would be dealt to CARDNAME.
- Prevent all damage that would be dealt to and dealt by CARDNAME.
- Prevent all damage that would be dealt by CARDNAME.
- Prevent all damage that would be dealt to CARDNAME.
- Protection from {type}
- Remove CARDNAME from your deck before playing if you're not playing for ante.
- Spells and abilities your opponents control can't cause you to sacrifice permanents.
- You can't pay life to cast spells or activate abilities.
- You can't sacrifice creatures to cast spells or activate abilities.
- You can't draw cards.
- You can't gain life.
- You can't lose the game.
- You can't win the game.
- You don't lose the game for having 0 or less life.
- You may choose not to untap CARDNAME during your untap step.
- You may have CARDNAME assign its combat damage as though it weren't blocked.
- Your life total can't change.

Developer Mode

[Forge_DevMode](#)

Remaining Cards

<https://docs.google.com/spreadsheet/ccc?key=0Aipjpk0ZcU8fdFIMczZRR2tmazZGSGZYeDh1Z24teVE&usp=sharing>

+ Add a custom footer