



Release: Frontiers Beta 2026-1 · Frontiers Tabletop Engine

Frontiers Tabletop Engine · frontiersengine.dev



FRO

Beta 20

Transferred from Frontiers' original host

The Beta releases were originally written and published on HackMD, Frontiers' first home, and were migrated here when the engine moved to its own site. Formatting, links, or images may occasionally show artifacts of that transfer. The rules text itself is preserved as released.

Frontiers is a **universal tabletop roleplaying game engine** designed to help creators build their own tabletop roleplaying games. It is not a complete game on its own.

Frontiers is built first and foremost as a design tool. It exists to remove the need to reinvent common mechanical solutions while still leaving room for creative expression. Whether you are building a fantasy adventure game, a science-fiction exploration system, or something entirely different, Frontiers is meant to serve as a baseplate to leap off.

At its core, the Frontiers Engine establishes the essential pillars of play: how characters act, how uncertainty is resolved, how progression works, and so on.

These systems are intentionally presented in a modular way. Designers can use the entire engine as written, selectively adopt specific components and modules, or modify the structure entirely with the basis of an underlying framework.

The engine is developed with an **open-source, collaborative philosophy**. Frontiers is expected to change, grow, and refine over time. New versions will be released as the engine evolves, but previous versions will always remain available for use.

Development Status

As of **February 13th, 2026**, the Frontiers Engine is in **active development**. The current available version is **Frontiers Beta 2026-1**.

Throughout this document, you will see sections marked with an  icon.

These sections link to expanded or supplementary documents that explore specific systems in greater depth. These linked documents are optional and exist to provide additional examples, guidance, or alternate approaches for designers who want them.

Frontiers is released under the Open RPG Creative (ORC) License. This allows you to use, modify, and distribute the engine, including for commercial projects, provided you comply with the ORC License and the attribution requirements outlined therein.

For full license terms and attribution, see the Complete License Document .

Additionally, Frontiers uses a lot of unique unicode icons to document important systems or mechanics, (  ), if they are not showing up for you, check out Fixing Missing Icons Guide. However, missing the icons will not change the actual usability of the rules and mechanics.

Frontiers resolves **♦Actions** and ***Activities** using the **Gradient Resolution System**, a framework designed to separate whether an Action or Activity succeeds from how that outcome manifests.

Rather than treating results as strictly binary, the Gradient Resolution System allows Actions and Activities to succeed or fail with varying degrees of effect. This gives the Game Master and players a shared language for expressing advantage and consequence without stalling play.

The engine involves **two resolution approaches** depending on the situation; both are considered core to the Frontiers Engine:

- **Basic Resolution**, which determines success or failure using a single Resolution Die.
- **Gradient Resolution**, which adds a Gradient Die to express the strength, cost, or benefit of a Resolution Die result.

Which approach is used is always a decision made by the table or by the system built on Frontiers. Some games may rely heavily on Gradient Resolution, while others may use it sparingly or only in specific situations. Both approaches are fully supported.

Dice

Frontiers uses two dice when resolving uncertainty:

Die	Name	Purpose
d20	Resolution Die	Determines whether an ♦Action or ★Activity succeeds
d10	Gradient Die	Determines how strongly that result manifests

The **Resolution Die** answers the question:

“Does the character succeed?”

The **Gradient Die** answers the question:

“What does that success or failure look like?”

When only success or failure matters, only the Resolution Die is rolled. The Gradient Die is introduced when the degree of outcome is meaningful to the situation.

Making a Resolution Roll

When a character attempts an Action or Activity with uncertainty, the following process is used.

First, the player declares their intent and describes what the character is trying to accomplish.

Then, the player then rolls the **Resolution Die (d20)** and applies the relevant **Attribute Modifier**. The result is compared against a target value.

Most challenges use a **Difficulty Rating (DR)** set by the Game Master. When two characters directly oppose one another, the roll is compared against another roll using **Contested Resolution (CR)**.

If the modified result meets or exceeds the target, the Action or Activity succeeds. Otherwise, it fails.

Afterwards, if the Action or Activity calls for **Gradient Resolution**, the player then rolls the **Gradient Die (d10)** to determine the intensity, consequence, or benefit of the outcome.

Interpreting Gradient Results

The Gradient Die expresses how far an outcome deviates from what would be considered a standard result.

By default, Gradient results fall into the following ranges:

Roll	Outcome
1–2	Major Negative deviation
3–4	Minor Negative deviation
5–6	Standard outcome, no deviation
7–8	Minor Positive deviation
9–10	Major Positive deviation

A **standard outcome** resolves the Action or Activity as expected, without additional benefit or complication.

Minor results introduce small shifts in the situation: a cost, a delay, an opportunity, or an advantage that affects what comes next. Major results represent turning points, breakthroughs, or serious complications that reshape the scene.

Importantly, a Gradient result does not override success or failure. A failed Action or Activity may still produce a useful effect, and a successful Action or Activity may still introduce a cost. The Gradient Die exists to keep the fiction moving forward, not to punish or reward arbitrarily.

Some systems built on Frontiers may prefer alternate Gradient interpretations, such as explicit critical successes and failures. These approaches are compatible with the engine but are not required.

Why Use Gradient Resolution?

Traditional d20 systems often treat results as binary: success or failure. This can cause scenes to stall when rolls fail repeatedly or succeed without consequence.

The Gradient Resolution System provides a way to express partial success, costly victories, and meaningful setbacks without abandoning clear mechanical resolution. It allows the fiction to move forward even when outcomes are uncertain or messy.

Designers and Game Masters decide how prominent Gradient Resolution is in their games. The engine provides the tool; how often and how strongly it is used is a matter of taste and design philosophy.

Difficulty Ratings (DR)

Most Actions or Activities are resolved against a **Difficulty Rating (DR)**, a target number representing how challenging the Action or Activity is under current circumstances.

The specific numeric range used for DRs depends on the system being built. Many Frontiers-based games use values between 5 and 25, but this is an example scale rather than a requirement. What matters is consistency within a system and clarity at the table.

Contested Resolution (CR)

When two characters directly oppose one another—such as in a race, grapple, or debate—both roll a Resolution Die and compare results.

The character with the higher modified result wins the contest. When the degree of victory or loss matters, one or both sides may also roll a Gradient Die to determine how decisive the outcome is.

Favor and Disfavor

Favor and Disfavor can modify any die roll in the game unless a rule explicitly states otherwise.

When a roll has Favor:

- Roll one additional die of the same type per stack of Favor.
- Use the highest single result among all rolled dice.

When a roll has Disfavor:

- Roll one additional die of the same type per stack of Disfavor.
- Use the lowest single result among all rolled dice.

Favor and Disfavor stack up to a maximum of 3 each.

Each stack of Favor cancels one stack of Disfavor on a 1-to-1 basis before dice are rolled.

Examples:

- 2 Favor and 1 Disfavor → 1 net Favor.
- 3 Favor and 3 Disfavor → Normal roll.
- 1 Disfavor only → Roll two dice, take the lower result.

Favor and Disfavor apply to the specific roll they modify and do not carry over to future rolls unless explicitly stated.

Frontiers is experienced through two primary modes: **Free Play** and **Encounter Play**.

These modes shift naturally as the story demands. Free Play supports open-ended, fiction-forward gameplay, while Encounter Play introduces structure when timing, pressure, and immediate consequences become important.

Understanding how these modes interact with ♦**Actions**, ***Activities**, and other aspects of play work is central to running and designing games using the Frontiers Engine.

Free Play

Free Play is the default state of Frontiers. Characters explore environments, converse with others, gather information, plan, and pursue goals without strict sequencing or turn order.

Time in Free Play is fluid. Minutes, hours, or longer stretches may pass quickly if the fiction allows.

In Free Play, characters primarily engage in ***Activities**.

A roll is only required when:

- The outcome is uncertain, and
- The result meaningfully impacts the narrative

If success is obvious or narratively irrelevant, the Activity succeeds automatically.

Free Play emphasizes character intent, pacing, and consequence over mechanical structure.

* **Activities**

An ***Activity** represents effort that does not require precise timing or structured resource tracking.

Activities are performed during Free Play. If a situation escalates into Encounter Play, Activities are no longer available unless explicitly stated.

Common Activities include:

- Investigation
- Crafting
- Travel
- Research
- Negotiation
- Rest and z^zDowntime

Activities may use the Gradient Resolution System when uncertainty matters.

Encounter Play

When timing becomes critical, danger rising, disaster unfolding, or stakes escalating, the game shifts into **Encounter Play**.

Encounter Play is used when a situation requires:

- Structured sequencing of **♦Actions**
- Measured positioning and distance
- Immediate consequences
- Resource tracking

Encounters are not limited to combat. They may represent chases, hazards, tense negotiations, or collapsing environments.

During Encounter Play, **★Activities** are unavailable, unless there is a rule-based exception.

Encounter Structure

Encounters track time using:

- **Rounds** — roughly 10 seconds of shared in-world time
- **Turns** — the window in which a participant acts

When order matters, participants act according to **✂Momentum**.

✂ Momentum

Each character has a starting **Momentum Value (MV)** ranging from 0 to 10.

Lower Momentum acts earlier.

Higher Momentum acts later.

At the start of an Encounter:

- A character may keep their Momentum Value (provided by Archetype), or
- Adjust it by adding or subtracting up to their **□ Finesse Modifier**

Momentum may be further adjusted situationally at the GM's discretion.

Acting later is not inherently worse. Certain Abilities or tactics benefit from reacting instead of initiating.

Some Abilities, effects, or environmental conditions may temporarily raise or lower MV beyond 0–10. This is permitted within the engine.

♦ Actions

An ♦**Action** is a discrete, mechanically significant effort performed during Encounter Play.

Unless stated otherwise, all ♦Actions cost **1 ♦Action Point (AP)**.

Common ♦Actions include:

Action	Description
Attack	Strike a foe with melee or ranged force
Activate	Trigger a power, ability, or extraordinary effect
Interact	Manipulate the environment: open, push, pull, drop, etc.
Utilize	Use tools, gear, or items carried
Improvise	Any situational ♦Action not covered above (GM adjudicated)

♦ Action Points (AP)

During Encounter Play, each participant begins their Turn with **3 Action Points (AP)**.

♦AP may be spent:

- During your Turn for Actions
- During the Round for Reactions

Action Points refresh to maximum at the start of each Turn.

♦ Free Actions

Some efforts are minor enough that they do not meaningfully tax time or focus. These are ♦**Free Actions**.

If an Ability costs a ♦Free Action, that Ability may be used **once per Round**. A character may use multiple different ♦Free Action Abilities in the same Round, but each one is still limited to once per Round, unless an Ability states otherwise.

♦Free Actions may only be used during the character's Turn unless an Ability explicitly states otherwise.

» Movement

Movement is handled separately from Actions.

» Movement Points (MP)

During Encounter Play, characters reposition using »**Movement Points (MP)**.

Base »MP Value = 3

Add your □ **Finesse Modifier** to determine total »MP.

1 »MP = 1 □Unit moved.

»Movement Points refresh at the start of each Turn and can be used anytime during a character's turn.

Movement does not normally cost ♦AP (some exceptions may exist).

□ Units

Distance in Frontiers is measured in □**Units**.

- 1 □Unit = 5 feet (1.5 meters)
- 1 □Unit = 1 square or hex on a tactical map

□Units are used for movement, ranges, and areas of effect.

↗ Energy

Some Actions require **greater exertion**, pushing beyond the routine.

These efforts are tracked using ↗ **Energy**, a limited resource representing stamina, exhaustion, and the will to keep going.

↗ Energy applies **in all modes of play**, but scarcity becomes most visible during Encounters, where recovery is limited and consequences mount.

Characters typically start with **3 ↗ Energy** and gain more as they advance in level, but depends on the system and its designers on how it is used more exhaustively outside of Abilities and Equipment.

↗ Energy is regained during z²Downtime at a rate of 1 Energy per hour.

Spending ↗ Energy

This is covered more deeply in the Abilities section of this Overview, but ↗ Energy spending is quite straightforward:

If an Ability with a level equal to the user's relative Archetype Level (or lower) is used, the user doesn't spend ↗ Energy.

If an Ability with a level higher than the user's relative Archetype Level is used, the user spends ↗ Energy equal to the difference (Ability level - Archetype level).

Overspending $\nless$ Energy

If a character does not have sufficient $\nless$ Energy to pay a required cost, they may Overspend.

When Overspending:

- The Ability or effect is declared normally.
- Immediately roll any even-sided die (default 1d20).
- On an odd result, gain 1 $\square$ Fatigue.
- On an even result, no additional penalty occurs.

Overspend allows $\nless$ Energy to drop below 0, up to the system's defined negative limit.

If the negative limit would be exceeded, the action cannot be taken.

Each instance of Overspend requires its own roll.

$\square$ Wounds, $\square$ Fatigue, and $\blacktriangle$ Downtime

$\square$ Wounds

$\square$ Wounds represent lasting physical trauma.

When a character is reduced to 0 $\clubsuit$ HP, they gain a $\square$ Wound and fall Unconscious, unable to take $\star$ Activities or $\blacklozenge$ Actions of any kind, or spend $\gg$ MP. They remain Unconscious until they regain at least 1 $\clubsuit$ HP. If the Encounter concludes, they automatically reawake with 1 $\clubsuit$ HP (granted, if an ally can reach them).

Characters can also gain wounds from significant damage taken or other effects in rare cases.

Maximum $\square$ Wounds Value = 1 + $\blacklozenge$ Vigor Modifier

Each $\square$ Wound inflicts a stacking:

- -2 to $\square$ Physical Defense Value
- -2 to $\square$ Mental Defense Value

If a character runs out of $\clubsuit$ HP and would exceed their maximum $\square$ Wounds, they die.

$\square$ Wounds are removed during $\blacktriangle$ Downtime at a rate of 1 per hour.

$\square$ Fatigue

$\square$ Fatigue represents mental exhaustion, stress, and overextension.

$\square$ Fatigue may be imposed by overspending $\nless$ Energy, failing to take $\blacktriangle$ Downtime, or other effects more commonly than gaining $\square$ Wounds, often from sources of mental stress or pressure.

Maximum $\square$ Fatigue Value = 1 + $\square$ Resolve Modifier

Each $\square$ Fatigue inflicts a stacking:

- -2 to $\square$ Awareness
- -2 to ∞ Intuition

If a character would exceed their maximum $\square$ Fatigue, they fall Unconscious, unable to take $\star$ Activities or $\blacklozenge$ Actions of any kind, or spend »MP, until they are no longer at or above their Maximum $\square$ Fatigue Value.

$\square$ Fatigue is removed during z^z Downtime at a rate of 1 per hour.

$\blacktriangle$ Downtime

z^z Downtime represents a period of rest, recovery, and reduced danger.

z^z Downtime may occur between Encounters, during travel, while resting in a secure location, or at the Game Master's discretion.

Each hour of uninterrupted z^z Downtime provides the following benefits:

- Regain 1 $\blacklozenge$ **Energy**
- Remove either **1 $\square$ Wound** (or) **1 $\square$ Fatigue**

A character cannot remove both a Wound and a Fatigue in the same hour.

If z^z Downtime is interrupted by danger, combat, or significant exertion, recovery for that hour is lost.

If a period of $\blacktriangle$ Downtime lasts at least 3 Hours, all characters that are taking Downtime recover 1/2 their Maximum $\oplus$ HP. This stacks.

Attributes

Every character in Frontiers is defined by four core Attributes:

- ◆ **Vigor**
- $\square$ **Finesse**
- $\square$ **Acuity**
- $\square$ **Resolve**

Attributes represent broad domains of capability. They define what a character is naturally suited for, how durable they are under pressure, and how effectively they perform when uncertainty matters. They determine effectiveness.

Each Attribute exists on a 0-10 scale.

Every Attribute produces two mechanical values:

- **Attribute Score** — the full 0-10 value

- **Attribute Modifier** — half the Score, rounded down, presented as "+1, +2, etc"

The **Score** is most commonly used to calculate totals, thresholds, and defensive values. The **Modifier** is most commonly added to Resolution Rolls, Abilities, and other active checks, which Attribute is used is often determined by the GM.

Derivatives

Derivatives are statistics shaped by Attributes, but function independently during play.

Some are influenced not only by Attributes, but also by Archetypes, Origins, Backgrounds, Equipment, and Abilities.

The primary Derivatives in Frontiers are:

⛑ **Health Pool (HP)**

☐ **Wounds**

☐ **Fatigue**

⚡ **Momentum**

⚡ **Energy**

☐ **Physical Defense**

☐ **Mental Defense**

☐ **Awareness**

∞ **Intuition**

🍷 **Carrying Capacity**

When an Attribute contributes to a Derivative, its section will explicitly state whether it uses the **Score** or the **Modifier**.

❖ **Vigor**

❖ Vigor represents physical strength, endurance, bodily resilience, and resistance to trauma.

It governs how much punishment a character can withstand, how long they remain standing after injury, and how physically demanding burdens affect them.

Vigor is commonly applied when brute force, stamina, durability, or survival under physical stress are involved.

Vigor influences three major areas: ⛑Health Pool scaling, 🍷Carrying Capacity, and Maximum ☐ Wounds. It also contributes directly to ☐Physical Defense.

⛑ **Health Pool (HP)**

⛑Health Pool represents a character's current physical vitality.

A character's starting ⛑Health Pool is determined by system design, typically using: (Origin Bonus + Background Bonus)

When a character Advances a Level (included level 1), their maximum $\oplus$ HP increases by:
($\diamond$ Vigor Modifier + Archetype Modifier)

Origin, Background, and Archetype determine baseline durability. $\diamond$ Vigor determines how efficiently that durability scales over time.

Carrying Capacity

 Carrying Capacity represents how much equipables, gear, and physical burden a character can manage without penalty.

A character's  Carrying Capacity is determined by system design, typically using a calculation like the following: (Base Value + $\diamond$ Vigor Modifier)

The intended starting value for  Carrying Capacity is often somewhere between 5-15, the final choice being up to System Designers and how they wish to integrate Equipment.

Maximum Wounds Value

 Wounds represent lasting physical trauma.

When a character is reduced to 0 $\oplus$ HP, they gain a  Wound and fall Unconscious, unable to take $\star$ Activities or $\blacklozenge$ Actions of any kind, or spend $\gg$ MP. They remain Unconscious until they regain at least 1 $\oplus$ HP. If the Encounter concludes, they automatically reawake with 1 $\oplus$ HP (granted, if an ally can reach them).

Characters can also gain wounds from significant damage taken or other effects in rare cases.

Maximum  Wounds Value = 1 + $\diamond$ Vigor Modifier

Each  Wound inflicts a stacking:

- -2 to  Physical Defense Value
- -2 to  Mental Defense Value

If a character runs out of $\oplus$ HP and would exceed their maximum  Wounds, they die.

 Wounds are removed during z^2 Downtime at a rate of 1 per hour.

Physical Defense

$\diamond$ Vigor contributes its **Score** to  Physical Defense.

 Physical Defense = $\diamond$ Vigor Score +  Finesse Score + Equipment Bonuses

Defenses act as a **Difficulty Rating (DR)** against incoming attacks.

When an attacker makes a Resolution Roll, they must roll **equal to or higher than** the target's relevant Defense to succeed.

Defenses also function passively. Some Abilities may declare a value against a specific Defense. If the target's Defense exceeds that value, the Ability's effect may be reduced or negated.

▢ Finesse

▢Finesse represents coordination, agility, reflexes, balance, and control of tempo.

It governs how efficiently a character moves, how precisely they act under pressure, and how effectively they react to rapidly shifting situations.

Finesse is commonly applied when speed, precision, positioning, dexterity, or reaction time are involved.

Finesse influences two major areas: »Movement and ⚡Momentum. It also contributes directly to ▢Physical Defense.

» Movement

»MP Value = 3 + ▢Finesse Modifier

1 »MP = 1 ▢Unit moved.

»Movement Points refresh at the start of each Turn and can be used anytime during a character's turn.

Movement does not normally cost ♦AP (some exceptions may exist).

⚡ Momentum

Each character has a starting **Momentum Value (MV)** ranging from 0 to 10.

Lower Momentum acts earlier.

Higher Momentum acts later.

At the start of an Encounter:

- A character may keep their Momentum Value (provided by Archetype), or
- Adjust it by adding or subtracting up to their ▢ **Finesse Modifier**

Momentum may be further adjusted situationally at the GM's discretion.

Acting later is not inherently worse. Certain Abilities or tactics benefit from reacting instead of initiating.

Some Abilities, effects, or environmental conditions may temporarily raise or lower MV beyond 0–10. This is permitted within the engine.

▢ Physical Defense

▢Finesse contributes its **Score** to ▢Physical Defense.

□Physical Defense = ♦Vigor Score + □Finesse Score + Equipment Bonuses

Defenses act as a **Difficulty Rating (DR)** against incoming attacks.

When an attacker makes a Resolution Roll, they must roll **equal to or higher than** the target's relevant Defense to succeed.

Defenses also function passively. Some Abilities may declare a value against a specific Defense. If the target's Defense exceeds that value, the Ability's effect may be reduced or negated.

□ Acuity

□Acuity represents perception, reasoning, adaptability, pattern recognition, and mental sharpness.

It governs how clearly a character observes the world, how quickly they process information, and how broadly they develop specialized capabilities.

Acuity is commonly applied when investigation, analysis, problem-solving, or environmental awareness are involved.

Acuity influences two major areas: □Awareness and additional Skills. It also contributes directly to □Mental Defense.

□ Awareness

□Awareness represents a character's perceptual acuity and attentiveness to their surroundings.

□Awareness is a rollable value equal to: □**Acuity Modifier**

Awareness is used when determining whether a character notices hidden threats, environmental changes, concealed objects, or subtle sensory details.

System Designers may allow Awareness to function passively in certain contexts, but it is fundamentally a roll-based derivative.

Skills

A character gains a number of additional **Skills** equal to their □**Acuity Modifier**.

These Skills represent expanded mental breadth, learned specialization, and adaptive capability beyond a character's baseline training.

□ Mental Defense

□Acuity contributes its **Score** to □Mental Defense.

□Mental Defense = □Acuity Score + □Resolve Score + Equipment or Ability Bonuses

Defenses act as a **Difficulty Rating (DR)** against incoming attacks.

When an attacker makes a Resolution Roll, they must roll **equal to or higher than** the target's relevant Defense to succeed.

Defenses also function passively. Some Abilities may declare a value against a specific Defense. If the target's Defense exceeds that value, the Ability's effect may be reduced or negated.

□ **Resolve**

□Resolve represents mental fortitude, emotional stability, composure under pressure, and psychological endurance.

It governs how much mental strain a character can endure before collapsing and how resilient they remain when subjected to stress, fear, or overextension.

Resolve is commonly applied when composure, conviction, morale, or resistance to psychological pressure are involved.

Resolve influences two major areas: ∞Intuition and Maximum □Fatigue. It also contributes directly to □Mental Defense.

∞ **Intuition**

∞Intuition represents a character's ability to interpret motives, emotional shifts, and underlying intent.

∞Intuition is a rollable value equal to:

□**Resolve Modifier**

It is used to assess deception, detect manipulation, read social dynamics, and interpret emotional nuance.

Resolve determines how stable and perceptive a character remains under social or psychological pressure.

□ **Maximum Fatigue Value**

□Fatigue represents mental exhaustion, strain, and overextension.

Maximum □Fatigue Value = 1 + □Resolve Modifier

Fatigue may be gained from:

- Overspending ≠Energy
- Extended lack of rest
- Severe psychological stress
- Narrative strain

Each □Fatigue inflicts a stacking:

- -2 to □Awareness
- -2 to ∞Intuition

If a character would exceed their maximum □Fatigue, they fall Unconscious and are unable to take ★Activities, ♦Actions, or spend »MP until their □Fatigue is reduced below its maximum.

□Fatigue is removed during z²Downtime at a rate of 1 per hour.

□ Mental Defense

□Resolve contributes its **Score** to □Mental Defense.

□Mental Defense = □Acuity Score + □Resolve Score + Equipment or Ability Bonuses

Defenses act as a **Difficulty Rating (DR)** against incoming attacks.

When an attacker makes a Resolution Roll, they must roll **equal to or higher than** the target's relevant Defense to succeed.

Defenses also function passively. Some Abilities may declare a value against a specific Defense. If the target's Defense exceeds that value, the Ability's effect may be reduced or negated.

Skills represent trained expertise, practiced knowledge, refined technique, disciplined study, or honed experience.

They reflect what a character has intentionally learned beyond natural aptitude.

A Skill does not determine whether a character may attempt an ♦Action or ★Activity.

It determines how effectively training enhances that attempt.

In Frontiers, Skills are binary: you either have them or you don't. This doesn't prevent someone from doing something without a Skill in it, it only determines whether or not they can use a Skill Benefit.

If the approach meaningfully involves training, specialized knowledge, or practiced execution, the GM may allow a relevant Skill to apply.

The **task itself** determines the Attribute.

The **approach** determines whether a Skill applies.

A character may apply only **one Skill Benefit per ♦Action or ★Activity**.

Skill Benefits

When a Skill applies to a roll, the player chooses **one** of the following benefits before rolling:

1. Favor

Gain Favor on either:

- The **Resolution Die (d20)**, or
- The **Gradient Die (d10)**

2. Enhanced Attribute

For this roll, **double your relevant Attribute Modifier**.

3. Narrative Edge

If the roll succeeds, gain a secondary narrative advantage appropriate to the Skill.

Examples may include: - Completing a task more quickly

- Securing advantageous positioning
- Reducing collateral damage
- Gaining leverage in negotiation
- Preserving resources

The GM approves the final scope of the benefit.

When Skills Apply

Skills apply when:

- The character's approach clearly relies on trained technique
- The character is leveraging specialized knowledge
- The character is performing something they have practiced deliberately

Skills do not replace Attributes.

They refine how Attributes are used.

If a character lacks a relevant Skill, they may still attempt the action using their Attribute normally, they simply do not gain a Skill Benefit.

Example Skill Categories

Each system built on Frontiers defines its own Skill list. Skills should reflect the tone, genre, and setting of the game.

Example categories include:

- **Athletics** — climbing, lifting, brute motion
- **Stealth** — moving quietly or unseen
- **Persuasion** — diplomacy, charm, negotiation
- **Deception** — misdirection, disguise, lying
- **Mechanics** — devices, repairs, bypassing locks
- **Medicine** — stabilization, treatment, diagnosis
- **Knowledge** — history, arcana, biology, culture
- **Survival** — navigation, tracking, foraging
- **Investigation** — pattern recognition, deduction
- **Performance** — artistry, distraction, morale

Designers determine:

- How broad or narrow Skills should be
- How Skills are learned
- Whether new Skills may be created during play

Skills are a major expression layer of character identity.

They define how a character distinguishes themselves beyond raw Attributes.

Competencies

Some capabilities are not Skills, but **Competencies**.

Competencies represent baseline proficiencies granted by:

- Archetypes
- Origins
- Backgrounds
- Equipment
- System design

A Competency determines whether a character can properly use certain tools, equipment, or complex systems without penalty.

Examples may include: - Weapon Competency

- Heavy Armor Competency

- Advanced Tool Use

- Magic Usage - Vehicle Operation

Competencies do not grant Skill Benefits.

They determine access, not enhancement.

A character without the required Competency may attempt use at the GM's discretion, but may suffer Disfavor, increased Difficulty, or narrative limitation.

It is advised to never merge Skills and Competencies unless your System is explicitly designed to deal with the balancing baggage of such a decision.

Abilities are defined techniques, powers, maneuvers, or special actions that give a character unique mechanical impact.

Attributes define capability.

Skills refine execution.

Abilities define what a character can do beyond the ordinary.

Every Ability in Frontiers follows a universal mechanical structure, regardless of how a specific system chooses to unlock or organize them.

Universal Ability Structure

Every Ability must define:

- **Ability Level**
- **Action Point Cost**
- **Energy Cost (if applicable)**
- **Effect**

The source of the Ability, Archetype, Origin, Background, Equipment, etc, is determined by system design.

The mechanical backbone remains consistent across all Frontiers-based systems.

Ability Levels

Every Ability has a **Level**, representing its complexity, potency, and scaling potential.

Ability Levels are compared to the character's relevant **Ability Provider Level** (typically Archetype Level).

Ability Level does not inherently equal Character Level.
It is compared against the level of the system feature granting it.

If an Ability's Level is **equal to or lower than** the provider's level, it may be used without Energy cost.

If an Ability's Level is **higher than** the provider's level, it requires $\leq$ Energy.

♦ Action Point Cost

Every Ability specifies how many ♦Actions are required to activate it.

An Ability may require:

- 1 ♦Action Point
- 2 ♦Action Points
- 3 ♦Action Points
- A ♦Free Action
- Or a combination

Action cost determines tactical weight during Encounter Play.

During Free Play, Action requirements translate into time, concentration, and narrative positioning, but still follow the same mechanical structure.

Reaction-specific Abilities will also cost ♦Action Points.

✂ Energy Cost

If an Ability's Level exceeds the relevant provider level, the Energy cost equals the difference.

Energy Cost = Ability Level – Provider Level

Ability Level Relative to Provider	✂ Energy Cost
Equal or lower	0
+1 Level	1
+2 Levels	2
+3 Levels	3
etc.	etc.

A character must have sufficient ✂Energy available to activate the Ability. If they overspend ✂Energy, normal Overspending rules apply.

Ability Sources

Frontiers does not dictate how Abilities are acquired.

Common implementation models include:

- **Archetype-based progression**

Abilities are granted automatically or selected as the Archetype levels up.

- **Talent trees or branching paths**

Abilities unlock based on prior selections.

- **Narrative milestones**

Abilities are gained through story progression.

- **Skill synergy systems**

Certain Skills unlock specific Abilities.

- **Equipment-bound abilities**

Powers granted by items or artifacts.

The engine only enforces:

- Level comparison for Energy cost
- Action requirement clarity
- Defined mechanical resolution

Designing Abilities

When designing Abilities for a Frontiers-based system:

- Keep Level meaningful — higher Level should imply stronger or broader impact.
- Keep Action cost proportional to effect.
- Use $\leq$ Energy as a pressure valve for burst potential.
- Avoid flat numerical inflation; prefer situational leverage.

Abilities should create identity, tempo shifts, and tactical choices, not simply larger numbers. In expanded documentation (clicking on the header of this section) you can find example abilities and more details on designing them.

Equipment represents physical tools, weapons, armor, devices, relics, or artifacts that characters can carry or wield.

Because Frontiers is an engine, not a genre-specific system, Equipment does not define what items do. Instead, it defines how items interact with:

- **Level**
- $\leq$ **Energy**
- **Competency**

Individual systems built on Frontiers determine what Equipment exists, what it does, and how it behaves mechanically.

Equipment Levels

All Equipment is be assigned a **Level**. All Equipment is considered Level 0 unless otherwise specified.

Designers are encouraged to explicitly label equipment Level for clarity, but unlabeled equipment defaults to Level 0 for all mechanical interactions.

An item's Level represents:

- Its potency
- Its complexity
- Its technological or magical advancement
- Its rarity or refinement

Equipment follows the same scaling philosophy as Abilities. If an item's Level is:

- **Equal to or lower than the character's Level** → it functions normally.
- **Higher than the character's Level** → it costs $\nearrow$ Energy equal to the difference.

How that strain manifests depends on whether the item is **Equipped** or **Activated**.

Competencies

Some Equipment requires a **Competency** to use effectively.

Competencies represent formal training, familiarity, or specialized knowledge.

They are typically granted by **Archetypes**, **Backgrounds**, or system-specific aspect.

If a character lacks the required Competency, the system designer may rule that the item:

- Cannot be used,
- Functions at reduced effectiveness,
- Or imposes additional penalties.

Equipped Items (Persistent $\nearrow$ Energy)

Equipped items are objects that provide continuous benefit while worn or wielded.

Examples may include:

- Armor
- Weapons
- Magical implements
- Technological enhancements
- Persistent relics or augmentations

If an Equipped item's **Level exceeds your Character Level**, reduce your **maximum $\nearrow$ Energy** by the difference while that item remains equipped.

Example:

- Level 2 character equips Level 4 armor.
- Difference = 2.
- Maximum $\nearrow$ Energy is reduced by 2 while worn, paid as an $\nearrow$ Energy Cost.

This reduction:

- Applies only while the item is equipped.
- Cannot reduce maximum $\nearrow$ Energy below 0 (if equipping would put the character below 0 Maximum Energy, they cannot equip it)

Activated Items (Costing $\nearrow$ Energy)

Activated items produce effects only when used.

Examples may include:

- Medical kits
- Grenades
- Spell scrolls
- Specialized devices
- Consumables or limited-use relics

If an Activated item's **Level exceeds your Character Level**, you must spend $\nearrow$ Energy equal to the difference when using it.

Example:

- Level 1 character uses a Level 3 device.
- Difference = 2.
- Cost = 2 $\nearrow$ Energy.

If the character lacks sufficient Energy, they may Overspend following the normal $\nearrow$ Energy rules.

Basic Gear

Simple tools, mundane gear, and common equipment may be assigned at **Level 0** or **without a level**.

Such items create no $\nearrow$ Energy strain and require no special scaling unless a system chooses otherwise.

Building With Equipment

When designing a system on top of Frontiers, consider:

- What items require Competencies?
- What items are persistent versus activated?
- How frequently should above-level equipment appear?
- How much $\nearrow$ Energy tension should equipment create?

Equipment is intended to introduce meaningful decisions:

- Do you wear the stronger armor and reduce your stamina?
- Do you conserve Energy or use powerful tools early?
- Do you specialize in Competencies to avoid strain?

Frontiers does not define what Equipment does.

It defines how Equipment interacts with **Level** and $\nearrow$ **Energy** — leaving genre, tone, and balance to the system designer.

Every character in Frontiers is constructed from three foundational elements:

- **Origin** — your fundamental nature
- **Background** — your formative experience
- **Archetype** — your path of growth

Origin

Origin represents what you fundamentally are.

It describes inherent nature, biological, cultural, constructed, mystical, or existential.

Origin answers:

- What is your physical or metaphysical form?
- Where do you come from?
- What innate traits shape you?

Origin is intentionally broad.

It may represent:

- Species or lineage

- Region or culture of birth
- Magical source or mutation
- Artificial construction
- Planar origin
- Societal caste or engineered purpose

Origin typically provides:

- **Base** ➕ **Health Pool contribution (often ranging from 1-10)**
- Minor Attribute influence (if desired by the system)
- **Innate** Abilities, traits, or features
- Thematic narrative grounding

Background

Background represents lived experience before the story begins.

It reflects training, hardship, profession, social role, and formative events.

Background answers:

- What did you practice?
- What shaped your worldview?
- What skills did you develop before becoming extraordinary?

Background may represent:

- Former professions (soldier, artisan, courier)
- Social positions (noble, exile, revolutionary)
- Academic or mystical study
- Survivor histories
- Criminal or underground experience

Background typically provides:

- **Base** ➕ **Health Pool contribution (often ranging from 1-10)**
- Starting Skills
- Equipment familiarity or Competencies
- Narrative leverage within specific contexts

Archetype

Archetype defines what the character actively becomes. It governs progression, specialization, and access to extraordinary capability.

In other systems these are most similar to "Classes".

Archetype answers:

- What role do you fill now?
- What defines your method of impact?
- How do you grow stronger?

Archetype most often determines:

- **Base + Health Pool scaling (read Vigor)**
- **Access to Abilities**
- Resource growth (↗Energy, etc.)
- Combat or narrative specialization
- Scaling features across levels

Each Archetype is intended to contain **10 levels**.

At every Archetype Level, the character should gain a meaningful benefit or improvement to their gameplay.

- New Abilities
- New Competencies
- Additional Skills
- etc

Advancement

Frontiers uses **Level-based progression**.

By default:

- Each Archetype contains 10 levels
- Characters may combine Archetypes or extend beyond 10 based on system design
- A soft cap of 20 total levels is recommended

Level advancement should feel significant.

Characters should not gain all progression benefits at once.

Frontiers supports structured milestone progression through **Vector Advancement**.

Vector Advancement

Characters progress through **Vector Marks**, representing focused growth decisions. **Upon gaining 3 Vector Marks, the character advances one Level.**

A Vector Mark cannot be selected more than once per level. Available Vector Marks:

- **Attribute Mark**

Increase two different Attributes by 1 (respecting Attribute caps).

- **Skill Mark**

Gain one additional Skill.

- **Ability Mark**

Gain one new Ability with Level $\leq$ (Provider Level + 2).

- **Energy Mark**

Increase maximum $\nearrow$ Energy by 1.

- **Additional Mark**

Gain a system-specific mechanical benefit decided by System Designers.

Starting Values (Default Recommendation)

At Level 1, characters typically begin with:

- **10 Attribute Points**

- **3 $\nearrow$ Energy**

- **3 Skills**

- **2 Abilities of Level 3 or lower**

Systems may modify these values or their restrictions to match tone, lethality, balancing, or genre expectations.

The rules presented in this Overview document define the **core structure** of the Frontiers Engine.

These systems, Attributes, Derivatives, Flow of Play, Skills, Abilities, Equipment, and Advancement, form the intended baseline framework for this version of Frontiers.

All values and structures are intentionally flexible, so feel free to modify, remove, adapt anything from Frontiers into your systems you wish to make.

This document represents the **standard ruleset** for Frontiers version: **Frontiers Beta 2026-1** (February 13th, 2026)

Modules

Below is updated list of released and in-progress modules for this Frontiers version:

Released

Coming Soon

Fantasy Foundations Module (by Jayke Paver) - Coming Soon **Sci-Fi Starter Module (by Jayke Paver)** - Coming Soon **Resources Expanded Module (by Jayke Paver)** - Coming Soon

Section Expansions (Further Rules)

Throughout this document, section headers marked with the  icon link to expanded documentation.

These expanded sections may include:

- Deeper mechanical clarification
- Optional rule variants
- Designer guidance and balancing discussion
- Example implementations
- Supplemental systems and modules

If you want additional detail or alternative approaches to a subsystem, begin by consulting its expanded entry.

Final Notes

Frontiers is an open framework.

It is designed to be **free, adaptable, and community-facing**.

The goal of this engine is to empower designers to build new systems, genres, and experiences without reinventing structural fundamentals each time.

Frontiers was originally developed as a shared toolkit, a foundation for building games across genres, and is released publicly to encourage experimentation, iteration, and expansion within the tabletop community.

Fans of Frontiers are encouraged to:

- Build original systems on top of this engine
- Create Modules and Systems
- Share improvements and refinements

Frontiers is released under the Open RPG Creative (ORC) License. This allows you to use, modify, and distribute the engine, including for commercial projects, provided you comply with the ORC License and the attribution requirements outlined therein.

For full license terms and attribution, see the Complete License Document.

If you are interested in contributing modules, collaborating on expansions, or providing structured feedback, send an email to "contact@jaykepaver.me"

Frontiers grows stronger through iteration and thoughtful design.

Thank you for building with it.