Terra Invicta Event Creation

Some number of events are drawn from a weighted deck every month; their dates and times are random over that month.

Events are set in a game object called a “template,” which is a set of static data. All kinds of game elements have templates and they cross-reference one another. All templates are identified internally by a unique string, called a dataName. Dynamic data (like current data about nations and regions) that goes in savegames are stored in gameStates, many of which are referenced by their template’s dataname. Localization strings also typically reference dataNames.

The following config values are set in TINarrativeEventTemplate.json.

dataName. Enter a friendlyName instead and the dataName will be built to proper style if the sheet is doing what it should.

friendlyName = Not used by code; just for you to help identify the template.

illustrationResource = The path to the illustration in its assetbundle if one has been completed. Leave blank if it hasn’t.

soundResource = As above, but for audio.

requiresAliens = This is for future use, put “true” if the event involves anything related to the alien invasion. If it’s something that could happen without it, put false.

Year/endYear = The first/last possible year this could happen. Only enter if there is some historical date we need to link the event to. Generally preferable to use reqTechDataName to trigger the event on technological progress.

earliestMonth/latestMonth = This is for seasonal events if no year is specified. If a year and a month are specified, it won’t happen before the implied date is hit.

reqTechDataName = The dataName of the global TITechTemplate that unlocks this event.

Publicity = (Global, Factions, Target). This is who is notified of the event after it is complete.

- Global = All factions. (
- Factions = All affected factions.
- Target = Only the faction / nation targeted by the event. Includes secondaries.
- Target_PrimaryOnly = Only the faction / nation targeted by the event.
- Actor = Only the faction who receives the pop-up to act on the event.
- Silent = Nobody sees any notification.

Repeatable = (Unlimited, OncePerFaction, OncePerCampaign)

- Unlimited = Can happen repeatedly with no limit.
- OncePerFaction = Each faction can encounter this event once. Use this if you've got a clever event that you don't want to be lost with the AI silently encountering it most of the time. Also use for something you want to target all factions simultaneously.
- OncePerCampaign = Happens exactly once.

numOptions = (1 to 4) How many choices the player has in this event. Max is four. The code will ignore options configured beyond this number.

forceEvent = This event will occur the first time the conditions are met for it to occur. Avoid with repeatable events.

reqEventUnlock = This event requires another event to occur before it can be triggered. This event is designated in the prior events options config; this is a simple Boolean to tell the game not to trigger it randomly until the prior event has occurred.

baseWeight = This is the likelihood of the event occurring; the total weights are summed and a % for each possible event is determined.

altBaseWeight/condition/ = TICondition … These four values designate a condition under which the game should use an alternate weight value. (APPENDIX has list of conditions and how to handle them)

> altBaseWeight/condition/$type = This is the class of a condition that tells the game to use an alternate weight value instead of baseWeight.
>
> altBaseWeight/condition/strIdx = Some conditions require a string input; it goes here.
>
> altBaseWeight/condition/sign = Conditions generally have a comparison; this is the sign of the comparison.
>
> altBaseWeight/value = This is the value we're testing the condition against.

monthlyWeightDelta = How much the weight changes each month from the current base/alternate weight. Note that negative numbers will take it below zero and effectively make the event impossible to trigger if it doesn't by a certain time.

weightDeltaWhenTriggered = How much the weight drops after the event is triggered. Irrelevant for non-repeating events.

targetType = (global, faction, nation, region, mapRegion, councilor, spaceBody, habSite, hab, fleet, ship) = This is what sort of game element is on the receiving end of the event.

- global = USE CAREFULLY. This should only be for one-off events with only one non-faction-specific choice. It only informs the human player what's happened as a performance-saving measure.
- faction = The event will target one or all qualifying factions.
- nation = The event will target one or all qualifying nations
- region = The event will target one or all qualifying regions. Regions are scenario-specific; mapRegions are universal.
- mapRegion = The event will target one or all qualifying regions associated with the mapRegions.
- councilor = The event will target one or all qualifying councilors.
- spaceBody = The event will target one or all qualifying space bodies (planet or asteroid)
- habSite = The event will target one or all qualifying hab sites.
- habs = The event will target one or all qualifying habs (stations and bases).
- fleet = The event will target one or all qualifying fleets.
- ship = The event will target one or all qualifying ships.
- priorActor = The event will target the faction targeted by a previous event.
- priorTarget = The event will target the GameState used as the target in a previous event.
- priorSecondary = The event will target the GameState used as the secondary in a previous event.

hitAllQualifyingTargets = If false (the usual setting), an event will only trigger for one qualifying target; this overrides that and hits everyone who meets the conditions to be a target.

cooldown_months = how many months can pass between this event triggering.

possibleTargetDataNames/ (open-ended array) = Only works for faction, nation, region, mapRegion, space body, and habSite. Use the template names to limit the list of possible candidates to these. This is generally for conditions we don't handle algorithmically with conditions, like a list of Earth regions where civilian gun ownership is particularly high.

targetConditions/ (open-ended array of TICondition config) = This list of conditions are used to filter down the candidate targets.

targetWeightModifiers/ (open-ended array of TICondition config) = This modifies how likely a single qualifying candidate target is to be selected from the list of candidates. (Otherwise all targets are equally likely). The "value" field is a multiplier (which is only relevant if multiple weight modifiers are used).

secondaryStateType = If an additional gameState needs to be involved in the event (typically a second nation or benefiting faction), it should be designated here. Should be careful that it is congruent with the event as being configured. Some of the allowed types are shown below; the full list can be found in TIEffectsState.cs.

- Nation_Ally: An ally of the target nation.

- Nation_NormalRelations: A nation with neither an alliance nor a rivalry to the target nation.
- Nation_Rival: A rival of the target nation.
- Nation_SmallerRival_NMF: A rival of the target nation with a smaller GDP and not being commanded by the faction addressing the event.
- Nation_WarEnemy: A nation currently at war with the target nation.
- Nation_Neighbor: A geographical neighbor of the target nation.
- Faction_AlienFaction: The alien faction.
- Faction_AlienProxy: The faction designated as the aliens proxy. (Usually the Servants).
- Faction_AlienAppeaser: The faction designated as the aliens' appeaser. (The Protectorate if present; otherwise the Servants.)
- InputState: A specific GameState, identified by dataName. No in-built validation, use with caution.
- PriorEvent_Actor: A faction targeted by a previous event.
- PriorEvent_Target: A GameState used as the target in a previous event.
- PriorEvent_SecondaryTarget: A GameState used as the secondary in a previous event.

secondaryStateConditions/ (open-ended array of TICondition config) = conditions to filter down the possible secondary states that can be selected.

**** This is the end of the config that sets up the event. Everything after this handles the options and outcomes of the event. ****

eventOptions structure (up to 4, number configured must equal eventOptions int above)

An "option" is a button that appears before the player. An "outcome" is the effects of an option that is selected.

IMPORTANT: if the event will impose costs on faction resources, the first option should always be the "Do Nothing" option and should impose no costs on the faction. The game will throw errors if a faction is charged a resource it doesn't have.

/condition – This is not available for the first option so one choice always appears and the game doesn't lock up. Additional options use TICondition config. Filtered conditions still appear to the player but cannot be selected. A use for this might be to limit a choice to a particular faction.

/baseAIPreference = base weight for the AI to select this option when facing this event

/useAIModifiers = List of modifiers from TIFactionState/AIValues to use to modify the AI's preference for the event (via multiplication).

- proAlien
- antiAlien
- protectHumanLife

- protectAlienLife
- dirtyTricks
- riskAversion
- protectCouncilors
- wantEarthWarCapability
- wantSpaceWarCapability
- wantSpaceFacilities
- wantPopularity
- gatherMoney
- gatherInfluence
- gatherOps
- gatherScience
- energyTechs
- materialsTechs
- informationTechs
- lifeTechs
- militaryTechs
- socialTechs
- spaceTechs
- preserveLife
- fleetSmalls
- fleetMediums

/outcome (open-ended List) = Outcomes are the results of an option. Each option can have multiple outcomes configured; if there is more than one, a die roll determines which one happens.

/weight = Governs the chance of this outcome occurring relative to the others.

/AIFavored = This is the outcome the AI wants to occur. Only matters if there is more than one.

/effectTemplateNames = (open-ended List) These are the dataNames of the effects that will trigger when this outcome occurs. See the TIEffectTemplate sheet.

/delayedEffectTemplateNames = (open-ended List) These are TIEffects that are only triggered after completing and closing the event. These are necessary for effects that potentially destroy GameStates, e.g. killing councillors or damaging armies.

/projectGrantedTemplateName = This outcome will grant a project for research by the faction with the configured dataName. See TIProjectTemplate sheet and feel free to propose new and additional projects.

/orgGrantedTemplateName = This outcome will grant a org to the faction with the entered dataName. See TIOrgTemplate sheet.

/addNarrativeEvent = This outcome will unlock the narrative event with the designated dataName. That event should have reqEventUnlock set to true.

/costbuilder = This imposes a resource cost on the faction if the outcome occurs. Negative numbers confer a one-time income.

TESTING

The file TINarrativeTemplate.json must be updated with build config for it to appear ingame.

In a game session, hit the tilde key to open the console. Enter triggerevent (dataName of event). Event conditions will be ignored in this instance.

LOCALIZATION

The loc file is TINarrativeTemplate.en.

eventDataName = enter the dataName of this event from config

X = enter the index of the option

Y = enter the index of the outcome, if multiple outcomes are configured for this option.

TINarrativeEventTemplate.displayName.*eventDataName* = The event headline, 3-4 words, no verb required, avoid text substitutions

TINarrativeEventTemplate.*eventDataName*.summary = Single sentence setting up the event; should be agnostic about the outcome. Up to about 15 words; the shorter the better.

TINarrativeEventTemplate.*eventDataName*.query = This is the full narrative that sets up the choice. 1-2 paragraphs.

TINarrativeEventTemplate.*eventDataName*.option*X* = Summarizing the decision. One short sentence. Appears in the button.

TINarrativeEventTemplate.*eventDataName*.optionDetail*X* = Explanatory text that appears in the tooltip. This is often not necessary; the game automatically adds text describing the mechanics-based outcomes (resource changes, effects triggering). If that's sufficient, use <skip/> tag to tell the game not to include this string.

IF THIS OPTION HAS ONLY ONE OUTCOME:

TINarrativeEventTemplate.*eventDataName*.optionResult*X* = This is the post-choice text that will appear in summaries of the event, including in the right-hand feed. Limit to 15 words at most

(and only if they are short words). You can substitute the summary text here with the {summary} tag.

IF THIS OPTION HAS MULTIPLE OUTCOMES:

TINarrativeEventTemplate.*eventDataName*.option*X*.outcome*Y*=Short summary of what this outcome involves, prior to making the selection.

TINarrativeEventTemplate.*eventDataName*.optionResult*X*.outcome*Y*.Summary = Short (again, 15 words or less) summary of the result.

TINarrativeEventTemplate.*eventDataName*.optionResult*X*.outcome*Y*.Detail = Longer text that will appear in a popup describing the outcome of the event.

Allowed text substitution tags

The full list of these can be found in TINarrativeEventTemplate.cs.

{eventDisplayName} = event displayName

{actorName} = The name of the game element that is making the decision, so the name of the nation or faction.

{actorNameWithArticle} = The actor name with a preceding article (typically “the”) if it takes one.

{actorNameWithArticleCap} = The actor with a preceding article, which is capitalized because it’s beginning a sentence.

{targetFactionName} = The name of the faction being called on to make the decision.

{targetNationName} = The name of the nation being called on to make the decision. (In gameplay terms, the faction running the nation will make the decision, but this will print the nation’s name)

{targetNationNameWithArticle} = The nation’s name, but with its article (if it has one).

{targetNationNameWithArticleCap} = The nation’s name, with its article capitalized.

{targetNationNameWithArticleAndPlacePrep} = The nation’s name, in the context of “We are in {targetNationName}”.

{targetRegionName} = The target region’s name (the largest city). Typically we write it as “the Dallas region” so the {targetRegionName} region.

{targetRegionBoostFacilityName} = The name of the space launch facility in the region.

{secondaryTargetNationName}, {secondaryTargetNationNameWithArticle}, {secondaryTargetNationNameWithArticleCap}" = The secondary nation's name in various forms.

{targetHabName} = The targeted hab.

{targetHabSiteName} = The targeted hab's site (if it's on a spacebody).

{targetNaturalSpaceObjectName} = The targeted hab's parent space body (including what a station is orbiting).

Templates

The following template is handy to quickly write out the text of the events. Simply copy and paste it into TINarrativeEvents.en (or equivalent for your local language), remove the entries you don't need, then mass-replace event_macro with the dataName of your new event.

TINarrativeEventTemplate.displayName.event_macro_short=
TINarrativeEventTemplate.event_macro_short.summary=
TINarrativeEventTemplate.event_macro_short.query=
TINarrativeEventTemplate.event_macro_short.option0=
TINarrativeEventTemplate.event_macro_short.optionDetail0=<skip/>
TINarrativeEventTemplate.event_macro_short.optionResult0=<skip/>
TINarrativeEventTemplate.event_macro_short.option1=
TINarrativeEventTemplate.event_macro_short.optionDetail1=<skip/>
TINarrativeEventTemplate.event_macro_short.optionResult1=<skip/>
TINarrativeEventTemplate.event_macro_short.option2=
TINarrativeEventTemplate.event_macro_short.optionDetail2=<skip/>
TINarrativeEventTemplate.event_macro_short.optionResult2=<skip/>
TINarrativeEventTemplate.event_macro_short.option3=
TINarrativeEventTemplate.event_macro_short.optionDetail3=<skip/>
TINarrativeEventTemplate.event_macro_short.optionResult3=<skip/>

TINarrativeEventTemplate.displayName.event_macro_long=
TINarrativeEventTemplate.event_macro_long.summary=
TINarrativeEventTemplate.event_macro_long.query=
TINarrativeEventTemplate.event_macro_long.option0=
TINarrativeEventTemplate.event_macro_long.optionDetail0=<skip/>
TINarrativeEventTemplate.event_macro_long.option0.outcome0=
TINarrativeEventTemplate.event_macro_long.option0.outcome0.Summary=<skip/>
TINarrativeEventTemplate.event_macro_long.option0.outcome0.Detail=
TINarrativeEventTemplate.event_macro_long.option0.outcome1=
TINarrativeEventTemplate.event_macro_long.option0.outcome1.Summary=<skip/>
TINarrativeEventTemplate.event_macro_long.option0.outcome1.Detail=
TINarrativeEventTemplate.event_macro_long.option0.outcome2=
TINarrativeEventTemplate.event_macro_long.option0.outcome2.Summary=<skip/>
TINarrativeEventTemplate.event_macro_long.option0.outcome2.Detail=
TINarrativeEventTemplate.event_macro_long.option1=
TINarrativeEventTemplate.event_macro_long.optionDetail1=<skip/>
TINarrativeEventTemplate.event_macro_long.option1.outcome0=
TINarrativeEventTemplate.event_macro_long.option1.outcome0.Summary=<skip/>
TINarrativeEventTemplate.event_macro_long.option1.outcome0.Detail=
TINarrativeEventTemplate.event_macro_long.option1.outcome1=
TINarrativeEventTemplate.event_macro_long.option1.outcome1.Summary=<skip/>
TINarrativeEventTemplate.event_macro_long.option1.outcome1.Detail=
TINarrativeEventTemplate.event_macro_long.option1.outcome2=
TINarrativeEventTemplate.event_macro_long.option1.outcome2.Summary=<skip/>
TINarrativeEventTemplate.event_macro_long.option1.outcome2.Detail=
TINarrativeEventTemplate.event_macro_long.option2=

TINarrativeEventTemplate.event_macro_long.optionDetail2=<skip/>
TINarrativeEventTemplate.event_macro_long.option2.outcome0=
TINarrativeEventTemplate.event_macro_long.option2.outcome0.Summary=<skip/>
TINarrativeEventTemplate.event_macro_long.option2.outcome0.Detail=
TINarrativeEventTemplate.event_macro_long.option2.outcome1=
TINarrativeEventTemplate.event_macro_long.option2.outcome1.Summary=<skip/>
TINarrativeEventTemplate.event_macro_long.option2.outcome1.Detail=
TINarrativeEventTemplate.event_macro_long.option2.outcome2=
TINarrativeEventTemplate.event_macro_long.option2.outcome2.Summary=<skip/>
TINarrativeEventTemplate.event_macro_long.option2.outcome2.Detail=
TINarrativeEventTemplate.event_macro_long.option3=
TINarrativeEventTemplate.event_macro_long.optionDetail3=<skip/>
TINarrativeEventTemplate.event_macro_long.option3.outcome0=
TINarrativeEventTemplate.event_macro_long.option3.outcome0.Summary=<skip/>
TINarrativeEventTemplate.event_macro_long.option3.outcome0.Detail=
TINarrativeEventTemplate.event_macro_long.option3.outcome1=
TINarrativeEventTemplate.event_macro_long.option3.outcome1.Summary=<skip/>
TINarrativeEventTemplate.event_macro_long.option3.outcome1.Detail=
TINarrativeEventTemplate.event_macro_long.option3.outcome2=
TINarrativeEventTemplate.event_macro_long.option3.outcome2.Summary=<skip/>
TINarrativeEventTemplate.event_macro_long.option3.outcome2.Detail=

Global text tags

<skip/> No string. (If a string is expected and not entered, the game will show the expected loc tag, which is an error. <skip/> overrides this.

 Line break. Style is to have

 at the end of a paragraph for readability.

<h></h> Highlight text, in a burnt orange color. Use sparingly.

Appendix

TIConditions, sample list. The full list of possible conditions is located in TICondition.cs.

- _f = compare to a float value.
- _i = compare to an integer value.
- _b = compare to a Boolean value.
- _e = include an enum of the relevant type in strValue field.
- _t = include a template dataName in the strValue field.

Signs

- EqualTo
- NotEqualTo
- GreaterThan
- GreaterThanOrEqualTo
- LessThan
- LessThanOrEqualTo
- Has (Contains)
- DoesNotHave
- OnlyHas

TINationCondition_fCohesion
TINationCondition_fDemocracy
TINationCondition_fEducation
TINationCondition_fInequality
TINationCondition_fMiltech
TINationCondition_fUnrest
TINationCondition_fPerCapitaGDP
TINationCondition_fChangePerCapitaGDP (in the last 30 days)
TINationCondition_iControlPoints
TINationCondition_iArmies
TINationCondition_iNavies
TINationCondition_iRivals
TINationCondition_iNuclearWeapons

TINationCondition_efPriority
TINationCondition_bFactionControlled
TINationCondition_bIsAlienNation
TINationCondition_efPublicOpinion

TIRegionCondition_bCoastal
TIRegionCondition_bOnTheWater
TIRegionCondition_eEnvironment
TIRegionCondition_eRugged
TIRegionCondition_fLatitude
TIRegionCondition_fAbsLatitude
TIRegionCondition_fLongitude
TIRegionCondition_bCoreEconomic
TIRegionCondition_bCoreResource
TIRegionCondition_fMonthlyBoost
TIRegionCondition_iMissionControl
TIRegionCondition_fXenoforming
TIRegionCondition_iAbductions :

TIFactionCondition_eIdeology
TIFactionCondition_bAlienProxy
TIFactionCondition_bAlienAppeaser
TIFactionCondition_efResourceValue
TIFactionCondition_iAtrocities
TIFactionCondition_tbHasEffect
TIFactionCondition_tbCompletedCampaignObjective
TIFactionCondition_tbUnlockedCampaignObjective
TIFactionCondition_tbCompletedMilestone
TIFactionCondition_tbCompletedProject
TIFactionCondition_bHumanFaction
TIFactionCondition_bCouncilorHasTrait
TIFactionCondition_iCouncilorStat

TIGlobalCondition_tbCompletedTech
TIGlobalCondition_fTemperatureAnomaly_C
TIGlobalCondition_iNuclearWeaponsUsed
TIGlobalCondition_bNuclearWeaponsUsed
TIGlobalCondition_bLooseNukes

TICouncilorCondition_bInHomeNation
TICouncilorCondition_bInFriendlyNation
TICouncilorCondition_bInRivalOrWarEnemyNation
TICouncilorCondition_bInWarEnemyNation
TICouncilorCondition_tbHasTrait

TICouncilorCondition_bInSpace