

Events (QSF)

Inhaltsverzeichnis

- [1 "Simple" medical incidents](#)
- [2 "Complex" medical incidents](#)
- [3 Simple fires](#)
 - [3.1 Generic building fire 01](#)
 - [3.2 Generic building fire 02](#)
 - [3.3 Generic building fire 03](#)
 - [3.4 Generic small object fire](#)
 - [3.5 Garbage fire](#)
 - [3.6 Generic forest fire 02](#)
 - [3.7 Generic vehicle fire 02](#)
 - [3.8 Generic water vehicle fire](#)
 - [3.9 Generic industrial fire with contamination](#)
- [4 Normal fires](#)
 - [4.1 Building explosion](#)
- [5 Gangster events](#)
 - [5.1 Pickpocketing](#)
 - [5.2 Robbery](#)
 - [5.3 Robbery at store](#)
 - [5.4 Arsonist 01](#)
 - [5.5 Arsonist 02](#)
 - [5.6 Drunk thug: Single perpetrator](#)
 - [5.7 Brawl](#)
 - [5.8 Car theft 01: Parking vehicle](#)
 - [5.9 Bomb planter](#)
 - [5.10 Escaped prisoner](#)
 - [5.11 Terrorists ignite dirty bomb](#)
- [6 Car accident](#)
 - [6.1 Car accident 01](#)
 - [6.2 Car accident 03](#)
 - [6.3 Car accident 04](#)
- [7 Other regular events](#)
 - [7.1 Defective switchbox](#)
 - [7.2 Burning switchbox](#)
 - [7.3 Building collapse](#)
 - [7.4 Earthquake](#)
 - [7.5 Missing person](#)
 - [7.6 Hunt for the murderer](#)
 - [7.7 Buried person](#)
- [8 World events](#)
 - [8.1 Munich, Hamburg, Berlin: Drunk person riots at beer tent](#)
 - [8.2 Munich, Hamburg, Berlin: Food poisoning](#)
 - [8.3 Munich , Hamburg, Berlin: Beer tent fire](#)

"Simple" medical incidents

- Patient has immediately health loss.
- Valid for all medical incidents: The injured person will be spawned at a special tagged spawn point.

Defined position of victim:

- Edit person spawn point on position.
- Assign the correct spawn pool in the spawn point [component](#).
- Attention: The spawn point can't be assigned to the layer named below, because it won't be found as long as the layer wasn't activated.
- Event tag: Property 'EventTag' in JSON.

Defined layer:

- Create a layer with an appropriate name.
- Contains decoration objects
- Layer has to be invisible in-game, don't forget to deactivate 'Load in game'
- The property in the JSON file is called 'EventLayer'

- [8.4 Munich: Bomb attack at Oktoberfest](#)
- [8.5 Hamburg: Stabber at fair](#)
- [8.6 Berlin: Chief of state kidnapped during visit at fair](#)

"Complex" medical incidents

- Patient is ill and will be injured from time to time.
- Event editing is the same as simple medical incident.

Simple fires

- You'll need a building with fire editing.
- For all fire [events](#): The target of the fire event has to be marked by a tag. Only this object will burn at the beginning.
- The forest fire is different, this fire event will be tagged by a meta entity with linked objects.

Defined position of fire object:

- Take a look at the event descriptions.
- Event tag: Property 'EventTag' in JSON - if not mentioned otherwise (for some [events](#) the 'DifficultyTags' will be used).

Generic building fire 01

- You'll need an object with building [component](#) and fire editing.
- ATTENTION: Bus station etc. shouldn't have this [component](#) - only real buildings are allowed to use the [component](#).
- Property in JSON in order to trigger the event: 'DifficultyTags' (one or more [events](#), separated by comma).

Generic building fire 02

- With person in the building
- You'll need an object with building [component](#) and object in event window slot and fire editing.
- Property in JSON in order to trigger the event: 'DifficultyTags' (one or more [events](#), separated by comma).

Generic building fire 03

- With person trapped in the building
- You'll need an object with building [component](#) and object in door slot and fire editing.
- Property in JSON in order to trigger the event: 'DifficultyTags' (one or more [events](#), separated by comma).

Generic small object fire

- You'll need an object with tree or hedge [component](#) and fire editing.
- Property in JSON in order to trigger the event: 'DifficultyTags' (one or more [events](#), separated by comma).

Garbage fire

- You'll need an object with event tag GARBAGE and fire editing.
- Property in JSON in order to trigger the event: 'EventTag'.

Generic forest fire 02

- With injured persons
- You'll have to create an invisible meta entity for every forest fire, this meta entity must have the EventMetaEntity [component](#).
- Link some objects (Trees, with fire editing) EventEntityID slot.
- Assign the tags FOREST_FIRE;FOREST_FIRE_P or FOREST_FIRE_MP fore the multiplayer mode.
- Add debug meshes to the EventEntityID slot. Every objects must have a spawn point for persons. 'Active' will be set to 'Off' and use 'forest_fire_person_event' as spawn pool.
- Add additional trees and spawn points for the multiplayer mode. You won't have to tag the objects (Trees, spawn points) separately. except for the multiplayer event. All objects will be used, both the non-tagged from the normal forest fire editing and with 'FOREST_FIRE_MP' tagged objects.

Singeplayer

Multiplayer

4-5 Trees

8-9 Trees

2-3 Spawn points

5-6 Spawn points

Generic vehicle fire 02

- The engine will choose a civil car with inactive or missing MoveToDespawn function and fire editing.
- Alternative: A parking civil car ('combined car') will be tagged with an event tag.
- Property in JSON in order to trigger the event: 'DifficultyTags' (one or more [events](#), separated by comma).

Generic water vehicle fire

- The engine will choose a vehicle with the tag 'BOAT' if fire editing is available.
- Only tag water vehicles with enough space around them for the motor boat. Big container ships get the additional tag 'CONTAINER_SHIP'.

- For each water vehicle on the map you'll have to edit four dummy meshes (container ships get eight) into the water beneath the ship.
- Place some dummy meshes onto the deck of the ship (two for normal and four for container ships).
- Pay attention to the space around the dummy objects, because the engine will spawn the persons there (rotation is also important).
- The dummy meshes get a spawn point [component](#), don't forget to deactivate the spawn function.
- The spawn points on the deck must have the tag 'ONDECK'.
- The spawn points will be linked into the DrowningSpawnPoints slot of the boat [component](#).
- At the start of the event, a random number of persons will be activated and set to drowning state.
- Attention: The spawn points in the water should be reachable by the motor boat. Place them with enough space to the ship, the shore and other objects.
- Attention: The water vehicles need fire editing and damage models/textures.
- Pool for spawn points: 'boat_fire_event'
- List of sailors:
 - em5/prefab/person/civil_sailor_male01
 - em5/prefab/person/civil_sailor_male02

Generic industrial fire with contamination

- The engine will choose a building with industrial plant [component](#).
- Link a contamination cloud into the CloudArray slot.
- A contamination cloud is a debug box with particle [component](#) and the ContaminationCloud [component](#). Scale the particle effects to the radius of the cloud.
- Set the delay and radius in the cloud [component](#).
- Create spawn points for the injured people and use the same 'DifficultyTag' for both the cloud and the building.

Normal fires

Building explosion

- The engine will choose an object with building [component](#) and fire editing.
- Every building needs an explosion particle effect. This particle effect will be linked into the fire particle layer of the building, just like fire particle effects. Activate the flag for explosions in the fire particle options (will be edited in the prefabs).
- Tag: EXPLOSION_BUILDING

Gangster [events](#)

List A = Prefabs of persons which are able to become gangsters

- civil_gangster_male01
- civil_hooligan_male01
- civil_hooligan_male02
- civil_rocker_male01
- civil_rocker_male02

List B = Prefabs of persons which are able to become persons running amok, arsonists or take part in brawls

- civil_gangster_male01
- civil_hooligan_male01
- civil_hooligan_male02
- civil_rocker_male01
- civil_rocker_male02

- civil_casual_male02
- civil_casual_male03
- civil_casual_male04
- civil_casual_male05
- civil_casual_male06
- civil_casual_male07
- civil_sportive_male01

List C = Prefabs of persons which are able to become victims of crime

- civil_business_female01
- civil_business_female02
- civil_business_male01
- civil_business_male02
- civil_casual_female01
- civil_casual_female02
- civil_casual_female03
- civil_casual_female04
- civil_casual_male02
- civil_casual_male03
- civil_casual_male04
- civil_casual_male05
- civil_casual_male06
- civil_casual_male07
- civil_sportive_female02

List D = Prefabs of persons which are able to become terrorists

- civil_terrorist_male01

List E = Prefabs of persons which are able to become escaped prisoners

- civil_convict_male01

Pickpocketing

- Random person from list A as gangster and list B as victim.

Defined position of the gangsters:

- Edit spawn points for the persons on suitable positions.
- Assign the suitable spawn pool (only ganster, see list A) in the spawn point [component](#).
- Define the event tag and assign it into the 'EventTag' property.
- Add the event tag [component](#) to the spawn point and assign the correct event tag.
- Attention: It's not the spawn tag ('Tag' in the spawn point [component](#) (this function isn't used in this case).

Defined positions of the victims:

- Edit spawn points for the persons on suitable positions.
- Assign the suitable spawn pool (only victims, see list B) in the spawn point [component](#).
- Us a separate event tag as for the gangsters, it's called 'VictimTag'.
- Define the event tag and assign it into the 'EventTag' property.
- Add the event tag [component](#) to the spawn point and assign the correct event tag.
- Attention: It's not the spawn tag ('Tag' in the spawn point [component](#) (this function isn't used in this case).

Defined position for fleeing of the gangsters:

- Place a despawn point on a suitable position or use existing ones.

- Add an event tag to the despawn point.
- Use a different event tag and not the ones for the gangster or victim.
- In the JSON file: Property is 'EscapeTargetTag'.

Robbery

- Random person from list A as gangster and list B as victim.
- Same settings as pickpocketing.

Robbery at store

- Random person from list A as gangster and list B as victim.
- The engine will choose a random person to become a gangster. The person will run into a building and leaves it with a suitcase and a gun. Then a randomly spawned civil person will leave the building, make a few steps as drunk / injured person and falls down to the ground.

Defined positions:

- Same settings as pickpocketing.
- You don't have to define a position for the person, because it will leave the building.

Defined positions of the building:

- Edit a building (shop) at a suitable position.
- The building must have a door.
- The building must have an event tag.
- JSON file: Property is "BuildingTag".

Arsonist 01

- No persons in the building.
- The arsonist will be chosen from list B.
- A random building with fire editing will be chosen and lit by the arsonist. After that, the arsonist will light the next building.

Arsonist 02

- With persons in the initial building.
- The arsonist will be chosen from list B.
- A random building with fire editing and an object in the event window slot will be chosen and lit by the arsonist. After that, the arsonist will light the next building.

Drunk thug: Single perpetrator

- A building with the tag 'BAR' will be chosen. A person from the list A will spawn from the building and the person will attack a random person from list B.

Defined positions:

- Same settings as pickpocketing.
- The difference is, that the gangster will spawn from a building with a tag defined by game designer.
- The building should be a pub or night club. Add the additional event tag in the editor. Don't delete existing tags, add them separated by comma instead.

Brawl

- A building with the tag 'BAR' will be chosen. One or more persons from list A will spawn and attack persons from list B nearby.
- Trespassing persons from list B can join the brawl.

Car theft 01: Parking vehicle

- The engine will choose a '_combined' vehicle with the tag 'CARTHEFT'.
- At the start of the event, the '_combined' vehicle will be replaced by a normal car (e.g. civil_car01_combined -> civil_Car01).
- In order to enable the car to find its way from the parking space to the street, you'll have to create an additional AI lane for the vehicle which was tagged with 'CARTHEFT'. The vehicle has to be placed right beneath the AI lane. Do not place the lane directly onto the vehicle, because AI lanes overwrite collisions.

Bomb planter

- Spread the bombs across the map, especially at easy to find locations and large buildings (Office_Old/Office_New/Highrise/Cinema etc.).
- The bomb must have the EM5 bomb [component](#) in the entity.
- TimeRunning = Time until the bomb explodes. If the time is running out, the bomb will explode. Even when the police is defusing it at this moment.
- TimeDefuse = Time it takes to defuse to bomb.
- If the bomb should explode, you'll have to add a fire / complex fire [component](#) to the object. You also can add particle effects by this way.

Defined positions:

- Same settings as pickpocketing.
- You don't need any victims.

Defined position of the bomb:

- Edit Debug_Box onto suitable location (see above).
- Edit event tag for the Debug_Box.
- JSON file: Property is 'BombTargetTag'.

Escaped prisoner

- Escaped prisoners will be spawned from list E.
- You should place many (20+) spawn points for terrorists / escaping prisoners on the map.
- You can use the same spots as for the incident 'Dirty bomb' and place additional spots in backyards.
- The spawn points should have the name 'JAILBREAK'.

Defined positions:

- Same settings as pickpocketing.
- You don't need any victims.

Terrorists ignite dirty bomb

- One or more persons will be chosen from list D.
- The engine will choose a building with the tag 'DIRTYBOMB' for the bomb explosion.
- These buildings must have the ContaminationComponent. This [component](#) has a cloud slot. In this slot you'll have to link particle effects and the cloud [component](#) (editing is the same as for the industrial fire).
- The buildings also should have fire editing for the explosion (same as gas explosion), you also can use the same buildings for this.
- If there are other targets than buildings, you'll also have to add explosions and the ObjectDamageComponent there.
- The explosion should occur in public places and not only at buildings.

Defined positions:

- Same settings as pickpocketing.
- You don't need any victims.

Defined position of the bombing target:

- Edit entity on a suitable position (see above)
- Add event tag to the entity.
- JSON file: Property is 'BombTargetTag'.

Car accident

- There aren't any dynamic accidents. The accidents have to be edited like in [EMERGENCY 4](#).

Car accident 01

- Every accident get it's own layer. If the event will be triggered, the engine will choose a random layer and activate it.
- The layer above has a predefined name: 'car_accident', the layers below contain the different accidents. The layers below get the name 'car_accident01', 'car_accident02' e.g.
- Injured persons will be placed in the layer and get the name 'CARCRASH_VICTIM'.
- The victims trapped in the cars will be linked into the ContainedEntities slot of the car. Every car can contain only one injured person. The clothes and style of the persons should match to the car.
- The position of the trapped persons is not important, they will be placed by the engine while being cutted out of the car. It depends on the door position of the damage prefab, from this position the fire fighter will make a few steps backward. The end position can be calculated, but isn't editable.
- Before the driver doors should be place for the movement of the fire fighters. The doors shouldn't be blocked by other cars or doors (there's a fallback for blocked doors, but this shouldn't happen too often).

Car accident 03

- With brawl.
- Every accident get it's own layer. If the event will be triggered, the engine will choose a random layer and activate it.
- The layer above has a predefined name: 'car_accident_brawl', the layers below contain the different accidents. The layers below get the name 'car_accident01', 'car_accident02' e.g.

- Editing is the same as for the regular car accident. Instead of injured or trapped persons the engine will spawn two thugs.

Car accident 04

- With contamination.
- Every accident get it's own layer. If the event will be triggered, the engine will choose a random layer and activate it.
- The layer above has a predefined name: 'car_accident_contamination', the layers below contain the different accidents. The layers below get the name 'car_accident01', 'car_accident02' e.g.
- Editing is the same as for the regular car accident. You'll have to add the additional 'civil_truck02_chemical_dmg'. The truck must contain ContaminationComponent.

Other regular events

Defective switchbox

- All suitable switchboxes(ebox_event prefab) get the tag 'EBOX_EVENT'. The 'ebox_event' prefab has the ebox [component](#). There are two slots in the ebox [component](#) for particle effects (flashes and sparks). This particle effects will be played in different intervals (with sounds).
- Attention: The 'ebox_event' prefab contains fire editing. The fire editing will be inhibited during the event by code, so the switchbox won't burn (the fire editing is for the event 'Burning switchbox'). The exbox should't have a damage [component](#) because the ebox [component](#) inherits from the damage [component](#).

Burning switchbox

- Occurs when the regular switchbox event isn't solved by the player in time. The 'ebox_event' objects starts to burn.

Building collapse

- In order to burry a person under an object, the burry [component](#)" must be assigned to the blocking entity (e.g. a rock) and the target person has to be linked into the BuriedEntities slot of the object.
- IsHidden: 'True', when the person should be invisible and should only be visible, when the object was removed.
- IsDragable: 'True', when the object should be movable by fire fighters.
- You should place 10-15 debries beneath a building on the map.
- Properties of debries:
 - Every debris prefab get it's own debrie mesh.
 - EventTag 'BUILDING_COLLAPSE_DEBRIS'.
 - 'debris_small' meshes are suitable.
 - Delete the regular mesh and add the mesh as debug mesh [component](#) (then the object will be insivible during regular gameplay).
 - Every object get it's own adult person as victim.
 - The person also needs a debug mesh. Choose any adult person (will be changed by code).
 - Attention: Do not rotate the persons in order to lay them down, this will be done by animation. Otherwise will be rotated two times. If you want to test it, just attach an animation (couldn't work with debug meshes).
 - Delete all [components](#) except [metadata](#), transform, pivot on ground placement und debug mesh.
 - Hard link with the debris (person is part of the debris prefab).
 - Disband from instantiation from prefab.
 - No burry [component](#) needed (in this case this will solved by the code).
- Properties of buildings:
 - EventTag 'BUILDING_COLLAPSE'.
 - The bulding needs meshes for all damage states.

- Place 4-6 special debris around the building.
- Place the debris with enough free space, so the rescue crane can reach any debris.
- Hard link the debris to the building.
- You can also add additional objects which will be shown after the collapse of the building. Add them to the 'CollapseRubble' slot of the building.

Earthquake

- No editing needed (the code injures persons randomly and sets buildings on fire, also car accident can be activated by code).

Missing person

- Place several spawn points on the map (debug boxes with EM5 spawn point [component](#) and the event tag 'MISSING_PERSON_SPAWN').

Defined position of victim:

- Edit spawn point for person on suitable position.
- Assign the suitable spawn pool in the spawn point [component](#).
- Event tag in JSON file: Property is 'EventTag'.

Defined position of murderer:

- Edit spawn point for murderer on suitable position.
- Edit an event tag into the spawn point.
- The event tag has to be different from the victim.
- JSON file: Property is 'MurderTag'.

Defined fleeing position of murderer:

- Edit despawn point for murderer on suitable position or use existing one.
- Edit an event tag into the despawn point.
- JSON file: Property is "EscapeTargetTag".

Hunt for the murderer

- See 'missing person'.

Buried person

Defined position for victim:

- Edit spawn point for person on suitable position.
- Assign the suitable spawn pool in the spawn point [component](#).
- JSON file: Property is 'EventTag'.

Defined layer:

- Edit layer with suitable name on the position.
- Contains 'decorations' objects.
- Contains 'bury' object, e.g. the stone or tree which covers the person.
- 'Bury' object needs bury [component](#).

- JSON file: Property is 'EventLayer'.

JSON file:

- Property 'Injury': Injury of the victim.
- Property 'EventTag': Tagged spawn point for the position of the victim.
- Property 'EventLayer': Layer which will be activated for the event, adds additional objects.
- Property 'IsHidden': Is the victim hidden or visible? If hidden, the victim has to be found by the rescue dog first.

World events

- Add a scaled debug box with the tag 'WE_AREA' in the layer 'Worldevent'.
- The area defined by the debug box is the area for the world event.

Munich, Hamburg, Berlin: Drunk person riots at beer tent

- Same list of persons as of 'brawl'.
- Event tag of the building: 'WE_BEERTENT'.
- The beer tents must be edited as building, they also need a door.

Munich, Hamburg, Berlin: Food poisoning

- Edit several snack bars which contain to the world event.
- The snack bars get the event tag 'WE_SNACK'.
- Place "em5/prefab/person/civil_clerk_male01" as snack bar vendor ('MoveToSpawnpoint' = off, event tag 'WE_SNACK_CLERK') and soft link the person to the snack bar.
- The event chooses several civil persons (except the snack bar vendor) which are in the event are 'WE_AREA'. The persons will walk to the attraction area of the 'WE_SNACK', stay there for a short period of time, get the 'MoveToSpawnpoint' back and fall down a short time later.

Munich , Hamburg, Berlin: Beer tent fire

- A building with the tag 'WE_BEERTENT' and fire editing will be lit.

Munich: Bomb attack at Oktoberfest

- Same as the bomb in the incident 'bomb planter'.
- The bomb should be placed in a hidden place beneath a beer tent. The area should be accessible by emergency units. If the bomb explodes, it should lit the beer tent and some objects around it.
- The gangster will be chosen from list A.

Defined positions:

- Same settings as pickpocketing.
- No victim needed.

Defined position of the bomb:

- Edit debug box on suitable position (see above).
- The debug box must have an event tag.
- JSON file: Property is 'BombSpotTag'.

Hamburg: Stabber at fair

- Will be chosen from list A and spawns from the building with event tag 'WE_BEERTENT'.
- Stabber list:
 - civil_rocker_male01
 - civil_rocker_male02

Berlin: Chief of state kidnapped during visit at fair

- This event is only available in Berlin.
- Create own layer 'we_hostagetaking'.
- Create a scaled debug box with the event tag 'WE_HOSTAGETAKING'. This will mark the area, where all civil persons and vehicles will be deleted in order to make place for the event.
- If there are emergency units or persons or vehicles taking part in the event in the area, the event won't start.
- Edit into the 'HOSTAGETAKING' area:
 - DebugMesh Politician (Start position + rotation) - Event tag 'WE_HOSTAGE'.
 - DebugMesh thug (Start position + rotation) - Event tag 'WE_HOSTAGETAKER01' - This thug stands nearby the politician and points his weapon at him.
 - DebugMesh thug (Start position + rotation) - Event tag 'WE_HOSTAGETAKER02' - This thug stands a few steps away and watches the area.
- Thug: List D
- List of politicians:
 - em5/prefab/person/civil_business_female01
 - em5/prefab/person/civil_business_male01