

```
<?xml version="1.0"?>

<config>
  <!-- Minimum and maximum kart versions that can be used by this binary.
        Older version will be ignored. -->
  <kart-version min="1" max="3"/>

  <!-- Minimum and maximum track versions that be be read by this binary.
        Older versions will be ignored. -->
  <track-version min="1" max="7"/>

  <!-- Minimum and maximum server versions that be be read by this binary.
        Older versions will be ignored. -->
  <server-version min="6" max="6"/>

  <!-- Maximum number of karts to be used at the same time. This limit
        can easily be increased, but some tracks might not have valid start
        positions for those additional karts. -->
  <karts max-number="20"/>

  <!-- Everything related to hardware configuration.
        url : The server used for reporting statistics to. -->
  <HardwareReportServer url="https://stats.supertuxkart.net" />

  <!-- Everything related to online play.
        url: The server used for online multiplayer.
        server-version : Version of the server API to use. -->
  <OnlineServer
    url="https://online.supertuxkart.net/api/"
    server-version="2"
  />

  <!-- Addon and news related settings
        url : The server used for addon.
        allow-news-redirects : If true we allow all the server urls to be redirected by the addons
news.xml file-->
  <AddonServer
    url="https://online.supertuxkart.net/dl/xml"
    allow-news-redirects="true"
  />

  <!-- Scores are the number of points given when the race ends. -->
  <grand-prix>
```

<!-- Establish the distribution of points in GP.

For a race of N karts ; the N-first point values are taken.

Then, they are sorted. E.g. ; 0 1 2 1 3 2 becomes 0 1 1 2 2 3.

Then these numbers are used to establish the DIFFERENCE of points between consecutive karts.

The smaller of the numbers is used to establish the score for the last kart and not the difference between 2 karts.

In the above example, the last kart will have 0 point, the one before before 1 (0+1) ; the one before 2 (0+1+1), the one before 4 (0+1+1+2), etc. until the 1st which have 9 (0+1+1+2+2+3)

There shall be at least as much points nodes as max-numbers kart -->

<points points="0" /> <!-- added with 1 kart, score for the last kart -->

<points points="1" /> <!-- added with 2 karts -->

<points points="1" /> <!-- added with 3 karts -->

<points points="2" /> <!-- added with 4 karts -->

<points points="2" /> <!-- added with 5 karts -->

<points points="1" /> <!-- added with 6 karts -->

<points points="3" /> <!-- added with 7 karts -->

<points points="1" /> <!-- added with 8 karts -->

<points points="3" /> <!-- added with 9 karts -->

<points points="4" /> <!-- added with 10 karts -->

<points points="1" /> <!-- added with 11 karts -->

<points points="2" /> <!-- added with 12 karts -->

<points points="5" /> <!-- added with 13 karts -->

<points points="1" /> <!-- added with 14 karts -->

<points points="3" /> <!-- added with 15 karts -->

<points points="6" /> <!-- added with 16 karts -->

<points points="4" /> <!-- added with 17 karts -->

<points points="2" /> <!-- added with 18 karts -->

<points points="7" /> <!-- added with 19 karts -->

<points points="1" /> <!-- added with 20 karts -->

<points points="8" /> <!-- added with 21 karts -->

<points points="1" /> <!-- added with 22 karts -->

<points points="2" /> <!-- added with 23 karts -->

<points points="1" /> <!-- added with 24 karts -->

<points points="9" /> <!-- added with 25 karts -->

<points points="4" /> <!-- added with 26 karts -->

<points points="1" /> <!-- added with 27 karts -->

<points points="10" /><!-- added with 28 karts -->

```
<points points="2" /> <!-- added with 29 karts -->
<points points="12" /><!-- added with 30 karts -->
</grand-prix>
```

<!-- Time in follow-the-leader after which karts are removed.

The last values applies for all remaining karts.

time-per-kart Additional time added to the interval

for each kart in the race. -->

```
<follow-the-leader intervals="30 20 10"
```

```
time-per-kart="0" />
```

<!-- Startup information.

Penalty: Penalty time if a kart accelerates before SET. -->

```
<startup penalty="1" />
```

<!-- How often a news message is going to be displayed before

it will be ignored. -->

```
<news max-display="10"/>
```

<!-- smooth-normals: If the normals (for wheel raycasts) should be smoothed.

This is a global setting, it still needs to be activated for each track individually.

smooth-angle-limits: If the angle between the normal of a vertex and

the normal of a triangle are larger than this value, the normal

of the triangle will be used in interpolating the normals. This

has the effect of an edge split modifier - if the flat track and (say)

a wall at a 90 degree angle are not separated, the normal at that

vertex will be 45 degrees, resulting in completely wrong physics.

The angle limit in this case will discard the 45 degrees, and use

the normal of the flat triangle (pointing upwards). In the worst

case (all three normals discarded, the interpolation will just

return the normal of the triangle (i.e. de facto no interpolation),

but it helps making smoothing much more useful without fixing tracks.

fps: The physics timestep size

default-track-friction: Default friction to be used for the track and

any track/library pbject.

default-moveable-friction: Default friction to be used for any moveable,

e.g. karts, bowling balls, ...

solver-iteation: Number of solver iterations. A lower number reduces

the quality, but can reduce bouncing effect.

solver-split-impulse:: by default bullet solves for velocity and

position at the same time, which can introduce bounce. Setting

this to 1 can reduce bounce.

solver-split-impulse-threshold: Penetration threshold for using split impulse (ignored if solver-split-impulse is false).

solver-mode: Bullet's solver mode is a bit mask, which can be modified.

This entry contains a space-separated list of mode-names to either set or unset in this bit mask. Any name starting with a '-' indicate that the bit is to be set to 0, otherwise the bit will be set.

This field takes two

values: the first value is 'and'ed with bullet's default values

(i.e. it can be used to unset bullet defaults), the second value

is 'or'ed (i.e. is used to set a bit). A value of -1 for 'and'

means to keep all bits. The valid names are listed in stk\_config.cpp

and correspond to the definitions in btContactSolverInfo.h, e.g.:

'randomized\_order' corresponds to the bit SOLVER\_RANDOMIZE\_ORDER.

-->

```
<physics smooth-normals="true"
  smooth-angle-limit="0.65"
  fps="120"
  default-track-friction="0.5"
  default-moveable-friction="0.5"
  solver-iterations="4"
  solver-split-impulse="true"
  solver-split-impulse-threshold="-0.00001"
  solver-mode=""/>
```

<!-- The title and default musics. -->

```
<music title="main_theme.music" default="kart_grand_prix.music"/>
```

<!-- Replay related values, mostly concerned with saving less data and using interpolation instead.

max-frames: Maximum number of transform events that can be saved in a replay/history file. With normal play, 900 are enough to store at least one minute, usually more.

delta-t Maximum time between saving consecutive transform events.

The recording will do more transform events when some kart data changes significantly.

delta-speed If the speed difference exceeds this delta, a new transform event is generated before maximum time.

delta-steering If the steering angle difference exceeds this delta, new transform event is generated before maximum time. -->

```
<replay max-frames="12000" delta-t="0.100" delta-speed="0.6"
  delta-steering="0.26" />
```

<!-- Determines the minimap related values.

size: The size of the minimap (scaled afterwards) 480 = full screen height)

ai-icon: The size of the icons for the AI karts on the minimap.

player-icon: The size of the icons for the player karts. -->

```
<minimap size="180.0" ai-icon="16.0" player-icon="20.0"/>
```

```
<urls donate="https://supertuxkart.net/Donate"
```

```
password-reset="https://online.supertuxkart.net/password-reset.php"
```

```
assets-download="https://github.com/supertuxkart/stk-assets-mobile/releases/download/">
```

<!-- Skidmark data: maximum number of skid marks, and

time for skidmarks to fade out. Maximum number will over

current number of karts, so the more karts, the less

skidmark is on track. -->

```
<skid-marks max-number="500" fadeout-time="60"/>
```

<!-- Defines when the upright constraint should be active, it's

disabled when the kart is more than this value from the track. -->

```
<near-ground distance="2"/>
```

<!-- How long the end animation will be shown. -->

```
<delay-finish time="1.0"/>
```

<!-- How long the music credits are shown. -->

```
<credits music="8"/>
```

<!-- time is the time till a bomb explodes. time-increase is the time added

to timer when bomb is passed on. -->

```
<bomb time="30.0" time-increase="0.0"/>
```

<!-- Powerup collect-mode decides what is collected if a kart has already an

powerup: same: get one more item of the same type.

new: always get a new item.

only-if-same: if the random item is the same one as the

one currently owned, increase the number, otherwise

no more/new items are given to the kart.

no-explosive-items-timeout determines if world time is less than this value

then no cake or basketball is given by bonus boxes. -->

```
<powerup collect-mode="new" no-explosive-items-timeout="15.0"/>
```

<!-- time: How long a switch is being effective.

items for each item list the index of the item it is switched with.

Order: giftbox, banana, big-nitro, small-nitro, bubble-gum, nolok-bubble-gum,

```
        easter egg-->
<switch time="5" items="1 0 4 4 2 2 6"/>
```

```
<!-- disappear-counter: How often bubblegum gets driven over before it disappears.
      shield-time: How long the bubblegum shield lasts
      restrict-weapons: If true, using weapons will destroy the user's shield -->
<bubblegum disappear-counter="1" shield-time="10.0" restrict-weapons="false"/>
```

```
<!-- explosion-impulse-objects is the impulse that pushes physical objects
      away if there is an explosion. -->
<explosion impulse-objects="500.0" />
```

```
<!-- Networking
      steering-reduction: Reduce a remote kart's steering by this factor
                          each frame. This helps reduce oversteering by high latency
                          clients when they only do minor steering adjustments.
      max-moveable-objects: Maximum number of moveable objects in a track
                          when networking is on. Objects will be hidden if total count is
                          larger than this value.
```

```
-->
<networking steering-reduction="1.0"
      max-moveable-objects="15"/>
```

```
<!-- The field of views for 1-4 player split screen. fov-3 is
      actually not used (since 3 player split screen uses the
      same layout as 4 player split screen) -->
<camera fov-1="80" fov-2="65" fov-3="50" fov-4="75" cutscene-fov="0.61" />
```

```
<!-- disable-while-unskid: Disable steering when stop skidding during
      the time it takes to adjust the physical body with the graphics.
      camera-follow-skid: If true the camera will stay behind the kart,
                          potentially making it easier to see where the kart is going to
                          after a skid. -->
<steer disable-while-unskid="false"
      camera-follow-skid="true" />
```

```
<!-- Default values for all karts
      ===== -->
```

```
<general-kart-defaults>
```

```
<!-- Skidding: increase: multiplicative increase of skidding factor in each frame.
      decrease: multiplicative decrease of skidding factor in each frame.
      max: maximum skidding factor = maximum increase of steering angle.
```

time-till-max: Time till maximum skidding is reached.

visual: Additional graphical rotation of kart. The graphical rotation of the kart also determines the direction the kart is driving to when skidding is stopped.

visual-time: How long it takes for the visual skid to reach maximum.

revert-visual-time: how long it takes when stopping a skid to revert the visual skid and bring visuals and physics in sync again.

angular-velocity: Angular velocity to be used for the kart when skidding.

min-speed: Minimum speed a kart must have before it can skid. Must be >0, otherwise the kart can skid at the start of the race.

time-till-bonus: How long a kart needs to skid in order to get a bonus.

bonus-force: A speedup applied to the kart which skidded for a while.

bonus-time: How long the bonus-force is applied.

bonus-force: Additional engine force (this is used to offset the fact that turning after skidding (e.g. to correct direction) often uses up the skid bonus).

post-skid-rotate-factor: a factor to be used to determine how much the chassis of a kart should rotate to match the graphical view. A factor of 1 is identical, a smaller factor will rotate the kart less (which might feel better).

physical-jump-time: Time for a physical jump at the beginning of a skid.

graphical-jump-time: Time for a graphics-only jump at the beginning of a skid.

reduce-turn-min/max: The steering done by the controller (which is in [-1,1]) is mapped to [reduce-turn-min, reduce-turn-max] when skidding is active (for left turn, right turn will use [-max, -min]). The effect is that while you skid (say left) you can adjust the direction of the turn the kart is doing somewhat by steering to the left and right, but you will always keep on doing a left turn, just more or less. -->

<!-- Kart-specific settings used by the AI.

use-slipstream: if the AI should try to overtake karts using slipstream.

disable-slipstream-usage: even if the AI is not trying to use slipstream, it can get a lot of bonus, esp. on easy since the AI creates trains.

Set this to true to make sure AI does not get any slipstream bonus.

shield-incoming-radius: Radius at which projectiles will be detected and trigger a shield usage.

false-start-probability: Probability of a false start.

min/max-start-delay: Minimum and maximum start delay.

See <http://www.humanbenchmark.com/tests/reactiontime/stats.php>

Average reaction time is around 0.215 s.

nitro-usage: Integer determining how well the AI uses nitro, from 0 to 4  
0 corresponds to no use ; while 1 to 4 corresponds to various degrees

of quality use (using it immediately for 1 to more context-aware strategies)  
item-skill: Integer determining how well the AI use items, from 0 to 5  
0 corresponds to no use ; 1 to use after a random time ; while 2 to 5 use more advanced tactics  
collect-avoid-items: if the AI should collect and avoid items, or just ignore them.  
handle-bomb: If the AI should actively try to pass on a bomb.  
skidding-threshold: only for old-style skidding: when sharp turn should be triggered. Smaller values means it will sharp turn earlier, resulting in better driving in tight curves.

max-item-angle: Items that would need more than this change in direction are not considered for collection.

time-full-steer is the time for the AI to go from neutral steering to extreme left (or right). This can be used to reduce 'shaking' of AI karts caused by changing steering direction too often. It also helps with making it easier to push the AI karts (otherwise micro-corrections make this nearly impossible). A value of  $1/\text{maxFPS} / 2$  will guarantee that the wheel can go from -1 to +1 steering in one frame, basically disabling this mechanism.

bad-item-closeness is the maximum distance between a good and a bad item which can force the AI to abandon a good item in order to avoid hitting a bad item. If the distance is larger, it is assumed that there will be enough time to change steering direction.

straight-length-for-zipper is the minimum length a straight section of the track should have in order to activate a zipper.

competitive when ahead of the player, or more competitive when behind the player.

skid-probability: Since the AI is usually very good at using skidding, this is used to implement some rubber-banding for the AI: depending on distance to the player, the AI will change the probability of skidding. This attributes takes a space-separated list of "distance:probability" pairs (negative distances meaning the kart is behind the player, a positive number that the AI is ahead of the player). Then list should have at least 2 entries. Depending on the actualy distance the kart has the probability is then linearly interpolated (if the AI is more than the largest

distance ahead, the value for the largest distance is used, and similarly if the kart is more than the minimum value behind).

first-speed-cap: Fraction of maximum speed the first AI kart should drive at. Used to slow down karts that are ahead of the player. Note that setting this to a value >1 does NOT increase the speed the kart can drive at!

last-speed-cap: same as first-speed-cap, but for last AI kart. AI karts in-between use an average of it with first-speed-cap.

collect-item-probability: Probability of the AI actually trying to collect an item (if an item is selected for collection in the first place).

-->

<ai>

```
<easy time-full-steer="0.1"
  straight-length-for-zipper="35"
  use-slipstream="false"
  disable-slipstream-usage="true"
  shield-incoming-radius="0"
  false-start-probability="0.08"
  min-start-delay="0.3" max-start-delay="0.5"
  nitro-usage="0"
  item-skill="1"
  collect-avoid-items="false"
  handle-bomb="false"
  first-speed-cap="-100:1.0 -50:0.9 0:0.85 100:0.65"
  last-speed-cap="-150:0.92 -50:0.75 50:0.6"
  max-item-angle="0.7" max-item-angle-high-speed="0.3"
  bad-item-closeness="6"
  collect-item-probability="0:0"
  rb-skid-probability="0:0.0"
  skidding-threshold="4.0"
/>
```

```
<medium time-full-steer="0.1"
  straight-length-for-zipper="35"
  use-slipstream="false"
  disable-slipstream-usage="false"
  shield-incoming-radius="10"
  false-start-probability="0.04"
  min-start-delay="0.25" max-start-delay="0.4"
  nitro-usage="1"
  item-skill="2"
  collect-avoid-items="true"
```

```
    handle-bomb="false"
    first-speed-cap="20:1.0 60:0.9 100:0.85"
    last-speed-cap="-50:0.94 0:0.85 100:0.75"
    max-item-angle="0.7" max-item-angle-high-speed="0.3"
    bad-item-closeness="6"
    collect-item-probability="-10:1.0 0:0"
    rb-skid-probability="0:0.0"
    skidding-threshold="3.0"
  />
<hard  time-full-steer="0.1"
  straight-length-for-zipper="35"
  use-slipstream="true"
  disable-slipstream-usage="false"
  shield-incoming-radius="8"
  false-start-probability="0.01"
  min-start-delay="0.15" max-start-delay="0.28"
  nitro-usage="2"
  item-skill="3"
  collect-avoid-items="true"
  handle-bomb="true"
  first-speed-cap="50:1.0 150:0.9"
  last-speed-cap="0:0.96 80:0.8"
  max-item-angle="0.7" max-item-angle-high-speed="0.3"
  bad-item-closeness="6"
  collect-item-probability="10:1.0 20:0"
  rb-skid-probability="-50:1.0 -20:0.7 20:0.2 50:0.0"
  skidding-threshold="2.0"
  />
<best  time-full-steer="0.1"
  straight-length-for-zipper="35"
  use-slipstream="true"
  disable-slipstream-usage="false"
  shield-incoming-radius="6"
  false-start-probability="0.0"
  min-start-delay="0.15" max-start-delay="0.2"
  nitro-usage="3"
  item-skill="4"
  collect-avoid-items="true"
  handle-bomb="true"
  first-speed-cap="0:1.0"
  last-speed-cap="0:1.0"
  max-item-angle="0.7" max-item-angle-high-speed="0.3"
  bad-item-closeness="6"
```

```
collect-item-probability="0:1.0"  
rb-skid-probability="0:1.0"  
skidding-threshold="2.0"  
</ai>
```

<!-- Parameters for the speed-weighted objects:

a bigger value for strength-factor leads to the speed of the kart more quickly affecting the strength of the animation (up to a maximum value that corresponds to the original animation) -->

```
<speed-weighted-objects strength-factor="0.05" speed-factor="1.0" texture-speed-x="0.0"  
texture-speed-y="0.0"/>
```

<!-- friction: slip used for bullet skidding. A high value (like 10000000) disables bullet skidding. -->

```
<friction slip="10000000"/>
```

<!-- collision

impulse-type: STK can apply an additional impulse in case of kart-track collision:

'none' : no additional impulse

'normal': impulse along the normal

'driveline': impulse towards the nearest driveline.

An impulse towards the driveline works nice when the kart is driving more or less correctly on the track - it pushes the kart in the right direction. But if the kart is significantly off track, it has severe problems (since an incorrect driveline point can be selected, pushing the kart in the wrong direction, sometimes even causing a 'zip-along-obstacle' effect.

impulse: an additional impulse to be applied in a non-frontal collision to push two karts away from each other.

impulse-time: The impulse will be applied over a certain time period, which results in less abrupt changes. If set to 0, the impulse is only applied once.

restitution: restitution value to be used for the kart rigid bodies.

The restitution used depends on the speed to avoid physics issues (a collision with high speed and high restitution will push the kart high up into the air). The values specified are speed:restitution pairs, the actual restitution will be interpolated based on the points specified here.

bevel-factor: for each point of the chassis collision box one additional point is added, resulting in a bevelled box shape.

The original Z coordinate of the chassis is multiplied by 1-bevelZ (i.e. the main box part of the shape is shortened). The bevel point has the original Z coordinate, and the X and Y coordinates of the box are multiplied with (1-bevelX) and (1-bevelY). A value of 0 for all bevel coordinates disables bevelling, and uses a simple box shape.

As an example, a value of 1 for x and z will result in a sharp 'arrow' like shape.

physical-wheel-position: Defines where the 'physical' (raycast) wheel will be located. It's a weight factor with 0 = being at the widest side of the bevel, 1 = at the front and narrowest part of the kart. If the value is less than 0, the old physics settings are used which places the raycast wheels outside of the chassis and results in more stable physical behaviour of the karts. -->

```
<collision impulse-type="normal"
    impulse="3000" impulse-time="0.1" terrain-impulse="160"
    restitution="0:1.0 5:1.0 20:0.2" bevel-factor="0.5 0.0 0.3"
    physical-wheel-position="0" />
```

<!-- Skidding: increase: multiplicative increase of skidding factor in each frame.

decrease: multiplicative decrease of skidding factor in each frame.

max: maximum skidding factor = maximum increase of steering angle.

time-till-max: Time till maximum skidding is reached.

visual: Additional graphical rotation of kart. The graphical rotation of the kart also determines the direction the kart is driving to when skidding is stopped.

visual-time: How long it takes for the visual skid to reach maximum.

revert-visual-time: how long it takes when stopping a skid to revert the visual skid and bring visuals and physics in sync again.

angular-velocity: Angular velocity to be used for the kart when skidding.

min-speed: Minimum speed a kart must have before it can skid. Must be >0, otherwise the kart can skid at the start of the race.

time-till-bonus: How long a kart needs to skid in order to get a bonus.

bonus-force: A speedup applied to the kart which skidded for a while.

bonus-time: How long the bonus-force is applied.

bonus-force: Additional engine force (this is used to offset the fact that turning after skidding (e.g. to correct direction) often uses up the skid bonus).

post-skid-rotate-factor: a factor to be used to determine how much the chassis of a kart should rotate to match the graphical view.

A factor of 1 is identical, a smaller factor will rotate the kart less (which might feel better).

physical-jump-time: Time for a physical jump at the beginning of a skid.  
graphical-jump-time: Time for a graphics-only jump at the beginning of a skid.

reduce-turn-min/max: The steering done by the controller (which is in [-1,1]) is mapped to [reduce-turn-min, reduce-turn-max] when skidding is active (for left turn, right turn will use [-max, -min]). The effect is that while you skid (say left) you can adjust the direction of the turn the kart is doing somewhat by steering to the left and right, but you will always keep on doing a left turn, just more or less. -->

```
<skid increase="1.05" decrease="0.95" max="2.5" time-till-max="0.5"
visual="1.25" visual-time="0.7" revert-visual-time="0.7"
min-speed="10" time-till-bonus="1.0 3.0"
bonus-speed="4.5 6.5" bonus-time="3.0 4.0"
bonus-force="250 350"
physical-jump-time="0" graphical-jump-time="0.4"
post-skid-rotate-factor="1"
reduce-turn-min="0.2" reduce-turn-max="0.8"/>
```

```
<kart-type>
  <light />
  <medium />
  <heavy />
</kart-type>
</general-kart-defaults>
```

<!-- Here are the default fonts file names for STK.

Please DO NOT report bugs if there're crashes when using your custom font. This usually happen because the character map of the font is not in unicode mode (let's take Chinese for example, some fonts of it store the characters in BIG5 mode, which leads to crash with STK), but the fonts are to blame, what's the point of not using industry standard nowadays...

-->

```
<fonts-list normal-ttf="Cantarell-Regular.otf wqy-microhei.ttf NotoSansThai-Regular.ttf
NotoSansHebrew-Regular.ttf NotoNaskhArabicUI-Regular.ttf"
digit-ttf="SigmarOne.otf"
color-emoji-ttf="NotoColorEmoji.ttf"/>
```

<!-- Maximum bones from all animated meshes in each frame to be uploaded for hardware skinning, For gles 3.0 the specification guarantees at least 2048, for TBO in desktop at least 65536 (max buffer size) / 64, SSBO at least 2^24 / 64, so 1024 will work everywhere. -->

```
<skinning max-bones="1024"/>
```

```

<!-- For users with libsquish:
  Use a slow but high quality colour compressor.
  kColourClusterFit = (32),
  Use a fast but low quality colour compressor.
  kColourRangeFit = (64),
  Use a very slow but very high quality colour compressor.
  kColourIterativeClusterFit = (256),
  STK default the low quality.
-->
<texture-compression quality="64"/>

<!-- List of default ports used, by default STK use random ports for client.
  The server discovery port has to be the same across all clients and servers.
-->
<network-ports server-discovery-port="2757" client-port="2758" server-port="2759"/>

<!-- Configurable values used in SmoothNetworkBody class:
  min-adjust-length and max-adjust-length: Error in length smaller or larger
  than this value will be ignored (ie show the real position).
  min-adjust-speed: Minimum linear velocity required by a body to have smoothing.
  max-adjust-time: Maximum time spent in each smoothing stage.
  adjust-length-threshold: The higher this value, the larger the correction curve.
-->
<network-smoothing min-adjust-length="0.05"
  max-adjust-length="4.0"
  min-adjust-speed="0.3"
  max-adjust-time="2.0"
  adjust-length-threshold="4.0"/>

<!-- List of network capabilities to handle different servers with same version.
-->
<network-capabilities>
  <capabilities name="report_player"/>
  <capabilities name="soccer_fixes"/>
  <capabilities name="ranking_changes"/>
</network-capabilities>
</config>

```