Add requirement vector "Tile" range

The "Local" range can currently be pretty confusing, since it contains some information on the current tile (e.g. terrain type and extras), but not all (e.g. different units on the tile). Since, depending on effect type, it's not viable to include all tile info in the "Local" range, that could be split off into its own "Tile" range.

Requirement types currently in "Local" that could be moved to "Tile" include (but aren't necessarily limited to): Extra, BaseFlag, RoadFlag, ExtraFlag, Terrain, Resource, MaxUnitsOnTile, TerrainClass, TerrainFlag, CityTile

Requirement types in "Local" that don't currently refer to the tile, but could be given such a meaning in the "Tile" range include (but aren't necessarily limited to): UnitType, UnitFlag, UnitClass, UnitClassFlag, UnitState, MinMoveFras, MinVeteran, MinHitPoints (referring to whether any unit on the tile has this)

Possible current use cases of this might include support units (as described here). Additionally, if there are ever to be any effect types specifically referring to an extra (e.g. to change its build or pillage time), those could refer to the extra at the "Local" range. This could be used to move extra build requirements from terrain.ruleset (where there can be only one requirement vector) to effects.ruleset (where there can be multiple), which would allow getting rid of roads' first_reqs.

Related issues:
Blocks Freeciv - Task #939772: S3_2 datafile format freeze (d3f) New

History

#1 - 2017-12-17 03:27 PM - Alina L.
A different suggestion thread just reminded me: Obviously, all the local-ranged requirement types that don't currently refer to the tile, but could be given such a meaning (e.g. UnitType) should also be included in Adjacent and CAdjacent ranges (and possibly some others that refer to tiles).

#2 - 2018-02-06 03:29 PM - Marko Lindqvist
- Sprint/Milestone set to 3.1.0

#3 - 2018-02-06 03:30 PM - Marko Lindqvist
- Blocks Task #673656: S3_1 datafile format freeze (d3f) added

#4 - 2019-03-25 02:34 PM - Lexxie L
This would really enable all kinds of fixes and workarounds to oppressive restrictions and inability to properly specify certain behaviours. I hope someone will do this!

#5 - 2021-10-18 12:25 AM - Marko Lindqvist
I've been delaying this practically since effects, requirements, and their ranges got introduced in freeciv-2.0 for the assumed slight performance hit from handling additional range for the only benefit of having slightly more readable interface. It's 202x now, and we're casually taking much bigger performance hits elsewhere, so that argument is not valid any more.
Still, not sure if we should get this to 3.1, i.e., to make this big a change to 3.1 ruleset formats at this point, or postpone to 3.2.

#6 - 2022-01-25 07:21 PM - Alina L.
I'm starting work on the first part of this – introducing the new Tile range and moving requirement types from Local to it as appropriate (i.e. when they're referring to the target tile). I'll raise a separate ticket for the followup re: unit-related requirements at Tile/(C)Adjacent range.

Marko Lindqvist wrote:
Still, not sure if we should get this to 3.1, i.e., to make this big a change to 3.1 ruleset formats at this point, or postpone to 3.2.

So, are we still targeting 3.1, or pushing this back? A quick git diff S3_1..master -G"REQ_RANGE_" suggests almost no relevant changes (in code, at least) between the two branches, so making patches for both will not be much additional work; i.e. on the implementation side, this seems to be
mainly a question of where to put rscompat stuff.

#7 - 2022-01-26 12:17 AM - Marko Lindqvist
- Sprint/Milestone changed from 3.1.0 to 3.2.0

Alina L. wrote:

Marko Lindqvist wrote:

Still, not sure if we should get this to 3.1, i.e., to make this big a change to 3.1 ruleset formats at this point, or postpone to 3.2.

So, are we still targeting 3.1, or pushing this back?

I would leave it to 3.2 at this point (trying to get S3_1 formats frozen in a couple of months). Nobody has disagreed in the 3 months since my previous comment.

#8 - 2022-01-26 12:18 AM - Marko Lindqvist
- Blocks deleted (Task #673656: S3_1 datafile format freeze (d3f))

#9 - 2022-01-26 12:18 AM - Marko Lindqvist
- Blocks Task #939772: S3_2 datafile format freeze (d3f) added

#10 - 2022-01-26 08:29 PM - Alina L.
- File 0001-Split-new-Tile-requirement-range-from-Local.patch added

Patch is attached. Migrated requirement types are Terrain, TerrainClass, TerrainAlter, CityTile, TerrainFlag, RoadFlag, Extra, MaxUnitsOnTile and ExtraFlag.

Notes:

• rssanity.c's sanity_check_req_vec and sanity_check_req_set functions tracked local_reqs_of_type – since these were only used for now-tile-ranged req types, I turned it completely into tile_reqs_of_type; if anything other changes made in parallel still need local, this will cause merge conflicts

• In the supplied rulesets, some requirements already used "Tile" or, in a handful of cases, "tile" ranges – these would quietly fall back to the default (i.e. Local, now Tile); I cleaned them up along the way

Followup ticket: OSDN#43704

#11 - 2022-01-26 11:14 PM - Marko Lindqvist

Could we avoid duplicating (and hardcoding) range names in rscompat_req_range_3_2(), i.e., to use their enum values instead?

Neither rscompat_req_range_3_2() or its call checks only if we are allowed to update a ruleset (compat_mode), but not if the ruleset loaded is already of 3.2 format. We should only convert those ranges when loading 3.1 ruleset, not when loading 3.2 ruleset. Granted it's currently harmless as there's no new meaning for "Local" range for any of those requirement types that we would need to preserve in 3.2 rulesets.

About testing: Are you aware of any reason why this could alter autogames, i.e., does it make sense to do regression testing for this with autogames? (running a game with the same seeds with and without the patch, and check that final savegames are identical, short of things like event cache time stamps)

#12 - 2022-01-26 11:36 PM - Alina L.

Marko Lindqvist wrote:

Could we avoid duplicating (and hardcoding) range names in rscompat_req_range_3_2(), i.e., to use their enum values instead?

Will do. Presumably this goes for the requirement / universal types as well?

Marko Lindqvist wrote:

Neither rscompat_req_range_3_2() or its call checks only if we are allowed to update a ruleset (compat_mode), but not if the ruleset loaded is already of 3.2 format. We should only convert those ranges when loading 3.1 ruleset, not when loading 3.2 ruleset. Granted it's currently harmless as there's no new meaning for "Local" range for any of those requirement types that we would need to preserve in 3.2 rulesets.

I thought about that too (and checked how S3_1 rscompat does it, which didn't help); the problem is that requirements are used in multiple different ruleset files – which file's version should be checked? All possibly relevant ones (and hope they don't disagree)? Just one of them (game? effects?)
and hope the others are the same? Infer it based on the name of the file passed to lookup_req_list() (which sounds bodgy and complicated)? Pass it as an additional parameter to lookup_req_list() (which seems like quite a significant interface change for something that won't be relevant most of the time)?

Marko Lindqvist wrote:

About testing: Are you aware of any reason why this could alter autogames, i.e., does it make sense to do regression testing for this with autogames? (running a game with the same seeds with and without the patch, and check that final savegames are identical, short of things like event cache time stamps)

I don't believe anything should change – if everything works as intended, this is purely a refactoring (from the perspective of the game).

#13 - 2022-01-27 03:17 AM - Marko Lindqvist

Alina L. wrote:

which file's version should be checked? All possibly relevant ones (and hope they don't disagree)?

Now that you mention that, I've long thought that an attempt to support individual files from different versions does not really make sense. There might be some theoretical use-case for it, but in practice mixed files would always indicate an error somewhere.

Opened https://osdn.net/projects/freeciv/ticket/43708 about what I think should be done to that general problem.

Now, should we consider that new ticket a dependency for this one, to be able resolve above issue cleanly, or do we have a good way to resolve it without?

#14 - 2022-01-27 03:41 AM - Marko Lindqvist

Alina L. wrote:

Marko Lindqvist wrote:

About testing: Are you aware of any reason why this could alter autogames, i.e., does it make sense to do regression testing for this with autogames? (running a game with the same seeds with and without the patch, and check that final savegames are identical, short of things like event cache time stamps)

I don't believe anything should change – if everything works as intended, this is purely a refactoring (from the perspective of the game).

Can you do such test runs with a couple of different rulesets, or do you want me to do it?

#15 - 2022-01-27 02:12 PM - Alina L.

- File 0001-Split-new-Tile-requirement-range-from-Local-v2.patch added

Marko Lindqvist wrote:

Now, should we consider that new ticket a dependency for this one, to be able resolve above issue cleanly, or do we have a good way to resolve it without?

While I generally prefer doing things cleanly, I'm not exactly keen on having a patch lying around (and periodically making sure no concurrent changes broke it) until the other patch is done and merged.

In the new patch, I changed it to check all relevant versions and only apply when all of them are old; i.e. soft-unsupporting mixed versions – if a ruleset author is using mixed versions, that's on them and they have to deal with it. It's not exactly a "good" way to resolve it, but it could be worse.

So this new patch could be used, but I also understand if you'd rather wait on the other one – I for one am glad not to be the one who has to make that final call.

#16 - 2022-01-27 02:14 PM - Alina L.

Marko Lindqvist wrote:

Alina L. wrote:

Marko Lindqvist wrote:

About testing: Are you aware of any reason why this could alter autogames, i.e., does it make sense to do regression testing for this with autogames? (running a game with the same seeds with and without the patch, and check that final savegames are identical, short of things like event cache time stamps)
I don't believe anything should change – if everything works as intended, this is purely a refactoring (from the perspective of the game).

Can you do such test runs with a couple of different rulesets, or do you want me to do it?

I'll give it a shot.

#17 - 2022-01-27 06:21 PM - Marko Lindqvist
Alina L. wrote:

I'm not exactly keen on having a patch lying around (and periodically making sure no concurrent changes broke it)

Yeah. This patch touches quite a many places, so it has relatively high risk of catching bitrot.

In the new patch, I changed it to check all relevant versions and only apply when all of them are old; i.e. soft-unsupporting mixed versions – if a ruleset author is using mixed versions, that's on them and they have to deal with it. It's not exactly a "good" way to resolve it, but it could be worse.

While I've not yet looked at this new version, my personal opinion is that mixed version support is already broken, and this patch is not increasing the breakage.

So this new patch could be used, but I also understand if you'd rather wait on the other one – I for one am glad not to be the one who has to make that final call.

As long as the thing has not been escalated to Maintainers group, i.e. remains within freeciv-dev, your vote has the same weigh as anybody else's.

#18 - 2022-01-27 06:21 PM - Marko Lindqvist
- Status changed from New to Resolved

#19 - 2022-01-27 06:41 PM - Alina L.
Alightly then.

Regression tests passed for each of the supplied (non-modpack) rulesets (I did one pair of 7-player, 250-turn games each).

#20 - 2022-01-28 04:53 AM - Marko Lindqvist
- File 0002-Split-new-Tile-requirement-range-from-Local.patch added

Marko Lindqvist wrote:

Alina L. wrote:

I'm not exactly keen on having a patch lying around (and periodically making sure no concurrent changes broke it)

Yeah. This patch touches quite a many places, so it has relatively high risk of catching bitrot.

Had happened already now that I tried to apply this to my own tree.

Rebased version attached. I also added network capstr bump, so there's one even if I forgot to update it when I push the commit (of course, that part bitrots immediately if there is any other protocol change going in)

#21 - 2022-01-29 02:31 AM - Lexzie L

Alina L. wrote:

A different suggestion thread just reminded me: Obviously, all the local-ranged requirement types that don't currently refer to the tile, but could be given such a meaning (e.g. UnitType) should also be included in Adjacent and CAdjacent ranges (and possibly some others that refer to tiles).

This is very exciting to remove some weird disabilities and handicaps that ruleset designers currently suffer. It feels VERY close to the #1 wishlist item of every ruleset designer, so forgive me if I bring it up...

Let's take the case of Civ1 and Civ2, but also this is a case that frustrates EVERY ruleset designer in similar ways...

Fortified units in cities don’t get the fortify bonus if there are City Walls but they DO get the fortify bonus if the attacker of the city ignores city walls
Currently, there is just no way to test reqs for the attacker type and the defender type in the same effects.

So if we split some stuff into Tile req, I would sure like to take part in another req called "Counterpart", it means the OTHER unit (attacker if defender, defender if attacker).

#22 - 2022-01-29 07:09 AM - Marko Lindqvist

- Status changed from Resolved to Closed
- Assignee set to Marko Lindqvist

Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001-Split-new-Tile-requirement-range-from-Local.patch</td>
<td>305 KB</td>
<td>2022-01-26</td>
<td>Alina L.</td>
</tr>
<tr>
<td>0001-Split-new-Tile-requirement-range-from-Local-v2.patch</td>
<td>306 KB</td>
<td>2022-01-27</td>
<td>Alina L.</td>
</tr>
<tr>
<td>0002-Split-new-Tile-requirement-range-from-Local.patch</td>
<td>306 KB</td>
<td>2022-01-28</td>
<td>Marko Lindqvist</td>
</tr>
</tbody>
</table>