License Guide for ThinkGeo 15

ThinkGeo 15 licenses **developers**, not machines. Each developer keeps a small personal key file. Every build checks it and embeds a signed token in the assembly it builds, and the built application needs nothing else to run, anywhere.

You are	Read
a developer building with ThinkGeo	For developers
setting up a build server, CI pipeline or Docker build	Build servers, CI and containers
upgrading from ThinkGeo 14	What changed from ThinkGeo 14
the person who manages your company's ThinkGeo account	For administrators

For Developers

How It Works

1. You keep a **license key** file in your user profile folder.
2. When you build (`dotnet build` or Visual Studio), a build step inside the `ThinkGeo.Core` package checks the key.
3. A valid license key lets the build pass and stamps a signed license token into your assembly.
4. At run time, ThinkGeo reads that token from your assemblies. Nothing has to be installed or activated on the machines you deploy to.

A license key covers one **product family**:

Family	Packages
Desktop	`ThinkGeo.UI.Wpf`, `ThinkGeo.UI.WinForms`
Web	`ThinkGeo.UI.Blazor`, `ThinkGeo.UI.WebApi`
Mobile	`ThinkGeo.UI.Maui`

The build checks the key against the ThinkGeo packages your project references. For example, a project that uses `ThinkGeo.UI.Wpf` needs a Desktop key.

Try ThinkGeo: The 30-Day Trial

No sign-up is needed to try ThinkGeo. The first build on a machine without a key starts a **30-day trial** on that machine:

- For 30 days the build succeeds. Warning `TG0001` says on which date the trial ends and how many days are left. Maps show a *Not Licensed for Runtime* watermark.
- From day 30 the build fails with error `TG0002` until a key is in place.
- Applications you built during the trial keep running, with the watermark.

To evaluate for longer than 30 days, contact sales@thinkgeo.com.

Get Your License Key

1. Sign in at helpdesk.thinkgeo.com and open **ThinkGeo 15 License Keys**.
2. The page lists all three product families. The ones your company owns carry their dates and a button; the others read *Not licensed*, with the address to ask about them. Press **Generate Key** on each family you work with, and it becomes **Download Key** once the key exists.



3. Save the files, with their names unchanged - `desktop.thinkgeokey`, `web.thinkgeokey`, `mobile.thinkgeokey` - in your profile folder:

	Folder
Windows	`%LOCALAPPDATA%\ThinkGeo\` (for example `C:\Users\you\AppData\Local\ThinkGeo\`)
macOS / Linux	`~/.thinkgeo/`

4. Build. A build with no `TG` line in its output means the key is in place. There is nothing to activate and nothing to restart, including Visual Studio.

Every `.thinkgeokey` file in the folder is read, so keep `desktop.thinkgeokey` and `web.thinkgeokey` side by side if you work on both. When a project uses several families, the build stamps one token per key.

File	What it is
`desktop.thinkgeokey`, `web.thinkgeokey`, `mobile.thinkgeokey`	your keys, one per product family
`nonmap.thinkgeokey`	a non-map key for every family; see below

The key belongs to you, not to the machine: use the same files on your laptop, your desktop and your virtual machines. Do not share a key file or commit one to a repository. Downloading a key again returns the same key. **Revoke** on the page replaces it with a new one.

If the page offers you no key, it says which of two reasons applies. *Your contact is not enabled for downloads* means your contact record is not marked as a developer: contact support@thinkgeo.com to be set up on your company account. *Every seat your company holds is taken* means the licenses are all in use: contact sales@thinkgeo.com to be given one of them, or to buy another. A single family that reads *No seat free* is the same thing for that family alone.

Non-Map License Key

If you build a solution that uses ThinkGeo but never work on its maps (a back-end service, or a colleague's UI you only compile), get a **non-map key** from the same page. Under it every ThinkGeo project builds, and maps render blank. Any developer set up for downloads on a company with a current subscription can get one, and it takes no seat.

Your License Key's Date and Renewals

Your key carries a date. What it limits depends on how you bought ThinkGeo:

- **Subscription:** the date is the end of your subscription. After it, every build stops with `TG0003`, whatever package version the project uses, until you renew and download your key again.
- **Perpetual license (legacy):** the date is the end of your Software Assurance. Every ThinkGeo package released **before** that date builds with your key, permanently. A package released after it fails with `TG0003` until you renew Software Assurance and download your key again.

In both cases, applications you already built and deployed keep running.

What the Build Tells You

Code	Severity	Meaning	What to do
`TG0001`	warning	No key found. The build succeeds while the machine's 30-day trial runs, and the message says how many days are left. Maps show a *Not Licensed for Runtime* watermark.	Put your key in place (Get your license key).
`TG0002`	error	No key, and the 30-day trial on this machine has ended. The build stops.	Put your key in place (Get your license key).
`TG0003`	error	Your key does not cover this build: your subscription has ended, or, with a perpetual license(legacy), the package was released after your Software Assurance date. It	Renew and download your key again. With a perpetual license you can also use a package released

Code	Severity	Meaning	What to do
		also appears when `ThinkGeo.Core` is not the official package.	before your Software Assurance date.
`TG0004`	error	The key is invalid or revoked, or it does not cover a ThinkGeo UI package your project references (for example a Web key in a WPF project).	Download your key again, get the key of the missing family, and delete old key files from the folder.
`TG0005`	message	Building with a non-map key: maps render blank.	Nothing.

These lines appear in the `dotnet build` output and in the Visual Studio Error List, together with the project name. The check runs before compilation and takes well under a second.

Deploying Your Application

An application built with your key behaves the same on your own machine and on every machine you deploy it to: the license travels inside the assembly. There is nothing to install, copy or activate on servers, desktops or devices, and trimming, single-file publishing and ReadyToRun keep it.

- **MAUI (Android, iOS, MacCatalyst, Windows)**: the key is used on the machine that builds the app; the device needs nothing.
- **Blazor WebAssembly**: the build is checked like any other project; the browser itself does not check the token.

Build Servers, CI and Containers

Only the machine that **compiles** needs a key. Build agents usually have no profile folder, so the key is passed in one environment variable:

Variable	Value
`THINKGEO_LICENSE`	Either the key text (the `tgk_...` line of a key file; several lines, one per family, are allowed), or the full path of a file that holds such lines.

The variable is read before the profile folder. A project that references a UI package needs a line of that family, as on a developer machine.

Which license key to use. Use the license key of a developer who holds a seat, or ask support@thinkgeo.com to set up a build account on your company and seat it. Store the key as a secret in your CI system. A key is revoked when its seat is taken back, so replace the secret when the developer it belongs to leaves.

Without a license key, every fresh build agent or container starts on day 0 of a trial and the build passes with warning `TG0001`. An application with a watermark can then leave the pipeline unnoticed. Add `-warnaserror:TG0001` to the build command to turn that into a failure.

GitHub Actions:

```yaml
- name: Build
  run: dotnet build -c Release -warnaserror:TG0001
  env:
    THINKGEO_LICENSE: ${{ secrets.THINKGEO_LICENSE }}
```

GitLab CI (define `THINKGEO_LICENSE` as a masked variable under Settings > CI/CD > Variables; for two families, paste both lines):

```yaml
build:
  script:
    - dotnet build -c Release -warnaserror:TG0001
```

Azure Pipelines (secret variables are not passed to scripts automatically, so map it):

```yaml
- script: dotnet build -c Release -warnaserror:TG0001
  env:
    THINKGEO_LICENSE: $(ThinkGeoLicense)
```

Docker, with a BuildKit secret: the key is mounted only for the build step and never written into an image layer. Here the variable holds the **path** of the mounted file:

```
FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build
WORKDIR /src
COPY . .
RUN --mount=type=secret,id=thinkgeo \
    THINKGEO_LICENSE=/run/secrets/thinkgeo dotnet publish -c Release -o /app
```

```
docker build --secret id=thinkgeo,src=$HOME/.thinkgeo/desktop.thinkgeokey -t
myapp .
```

A shell on a build server:

```
export THINKGEO_LICENSE="$(cat ~/.thinkgeo/web.thinkgeokey)"   # or the path of
the file itself
dotnet build -c Release
```

What Changed from ThinkGeo 14

	ThinkGeo 14	ThinkGeo 15
Development license	Product Center installs a machine-bound license per product	A personal key file per product family, from the help desk; it works on every machine you use
When it is checked	At run time, when a debugger is attached	At **build time**, in `dotnet build`, Visual Studio and CI; the build output says what is wrong
Trying the product	Register, install Product Center, start an evaluation	Automatic: 30 days from the first build on a machine, with no sign-up
Build servers	Nothing needed: builds were not checked	A license key in the environment variable `THINKGEO_LICENSE`; without one, the agent builds on a 30-day trial
Deployed applications	Need a runtime license file	Need nothing: the build embeds a signed token in your assembly

	ThinkGeo 14	ThinkGeo 15
Mobile apps (MAUI)	Load a `.mapsuitelicense` file with `LicenseLoader.LoadLicense(...)` at startup	Nothing in code. Remove the call and the `.mapsuitelicense` files; the call is obsolete (a build warning now, an error in a later release)
Non-map builds	A runtime license on the developer's machine	A non-map key builds any ThinkGeo project with blank maps

ThinkGeo 14 licenses do not carry over: development licenses installed by Product Center and runtime license files do not license ThinkGeo 15 packages. A ThinkGeo 15 application needs a license key when it is built. Applications that stay on ThinkGeo 14 packages keep using their ThinkGeo 14 licenses as before.

Your purchases and renewals are unchanged; the help desk derives the seats from them.

To upgrade a team:

1. Check that every developer who needs a key is set up for downloads on your ThinkGeo account (see For administrators). Developers who had *Enable My Download* in ThinkGeo 14 already are.
2. Each developer downloads their keys; build servers get `THINKGEO_LICENSE`.
3. Move to the ThinkGeo 15 packages. In MAUI apps, remove `LicenseLoader.LoadLicense(...)` and the `.mapsuitelicense` files.

For Administrators

There is nothing to administer on the help desk: the main contact opens the same **ThinkGeo 15 License Keys** page as every other developer. Seats follow your purchases, and your developers take them by downloading. What needs a decision, who counts as a developer and whose seat comes back, ThinkGeo does for you on request.

Seats

- Each purchase gives seats to a product family: ten Desktop licenses are ten Desktop seats. These numbers follow *Order History*.
- A developer's key takes one seat of its family; each person holds one key per family. Downloading the key again does not take another seat.
- The page shows each family you own and the date it is valid until. A non-map key takes no seat.

Who Can Take a Seat

- Only a contact marked as a developer on your ThinkGeo account can download a key at all. ThinkGeo sets that mark: contact support@thinkgeo.com to have someone added or removed.
- Developers who had *Enable My Download* in ThinkGeo 14 are already marked.
- Those developers take the seats by downloading, first come first served. Nobody is seated in advance, and a developer who downloads nothing uses none.
- When every seat of a family is in use, the next developer is told so on their page and pointed at sales@thinkgeo.com. ThinkGeo can free a seat, or pin one to a particular person so that it is waiting for them; taking a seat back revokes the key held of it.

Typical Situations

Situation	What to do
A developer joins	They register on the help desk with their company e-mail address, and you ask support@thinkgeo.com to set them up as a developer. They then download the families they work on.
A developer leaves	Ask support@thinkgeo.com to take their seats back. Their keys are revoked, the next ThinkGeo release refuses them, and the seats are free again. Replace any CI secret that held their key.
A key may have leaked	The developer presses **Revoke** on their own key and downloads the new one. Nothing is needed from you.
You renewed	Developers download their keys again to pick up the new date, and you replace the key in any CI secret. With a subscription, builds stop after the old end date

Situation	What to do
	until the new key is in place. With a perpetual license, builds against packages released before the old date keep working in the meantime.
You need more seats	Contact sales@thinkgeo.com.

Questions

Can I use my key on my laptop and my desktop? Yes. The key belongs to you, not to the machine.

I lost the key file. Download it again; it is the same key.

Someone may have seen my key. Press **Revoke** on the help desk and download the new key.

My subscription ended. Builds stop with `TG0003` until you renew and download your key again. Deployed applications keep running.

My Software Assurance ended. Packages released before that date keep building, permanently. Newer packages show `TG0003` until you renew. Deployed applications keep running.

Can I skip the check? Setting `ExcludeAssets="build;buildTransitive"` on the `ThinkGeo.Core` package reference skips it. Maps then show the watermark, so this does not license anything.

Where do I get help? Contact sales@thinkgeo.com about seats and renewals, and helpdesk.thinkgeo.com about everything else. Include the exact `TG` line from your build output.