Solution Settings Reference

Configure global solution properties.

The Solution Settings page provides centralized configuration for solution-wide parameters including platform targeting, licensing, runtime options, and interpreter settings.

Access: Solution Settings

Settings Sections

Solution Information

Read-only information about the current solution:

Field	Description	Example
Name	Solution filename	MyProject
File Size	Database size	45.2 MB
Description	User-defined text	Production monitoring system
Solution Path	Full file location	C:\Solutions\MyProject.dbsln
Tag Count	Local tags	1,234
External Tags	From TagProviders	567
Help URL	Custom documentation	https://docs.company.com/

Target Platform and License

Core deployment configuration:

Setting	Options	Impact
Target Platform	Windows Only / Multiplatform	Framework and OS support
Product Family	FrameworX / FactoryStudio / EdgeMachine / EdgeConnect	Feature set
Product Model	150 / 300 / 500 / 1500 / 5000 / Unlimited	Point limits

Solution Options

Solution-wide options:

Option	Description	Default
Culture Info	Language/locale	en-US
Allow Client Tasks	Designer editing option. When checked, the Designer lets you set a Task's domain to Client and shows the built-in client tasks. It does not change runtime behavior: the WPF Rich Client runs its client scripts, including ClientStartup, ClientShutdown, and Client-domain classes, whether or not the option is checked.	Unchecked
Diagnostics Location	Log file path	Local

Retentive Database Settings

Click gear icon to configure:

Field	Description	Options
Provider	Database type	SQLite, SQL Server, PostgreSQL
Database	Database name	Solution.dbRetentive
ServerIP	Remote server	localhost or IP
Password	Authentication	Optional

Run Modules Inside Main Process

Controls whether each module runs inside the main TServer process or in a dedicated isolated process (`TProcessDomain.exe`).

Module	Checkbox label	Default
Script Tasks	Script Tasks	Unchecked
Device	Device Channel(s)	Unchecked



Both settings default to disabled (unchecked). Leave them disabled unless the hardware constraints described below apply.

When to enable

Enable only when **all three** of the following conditions are true:

- The target hardware has **severely limited CPU** (e.g., embedded or edge controllers).
- The target hardware has **severely limited memory**.
- The solution contains **very few modules/channels** (typically one or two).

This is a last-resort optimization for resource-constrained deployments. For any normal deployment, leave both disabled.

Trade-offs

	Out-of-process (default)	In-process (enabled)
Process isolation	Full isolation — a module crash does not affect TServer	No isolation — a module fault can destabilize the entire runtime
Security boundary	Module code runs in a separate security domain	Module code runs inside the main process
Performance	Slight inter-process overhead	Marginal reduction in IPC overhead
Module synchronization	Cross-process synchronization	In-process synchronization (marginally more efficient)

Diagnostic: confirming the execution mode at runtime

In the Devices runtime monitor, each channel displays:

- **Process Name** — shows `TServer` when the channel runs inside the main process; shows `TRunModule` when running in a separate process.
- **Process PID** — the PID of the hosting process.

Python Interpreter

Python integration paths:

Path Type	Description	Example
Local	Client machine	C:\Python39
Server	Server machine	/usr/bin/python3

Global Script Namespace Declarations

From FrameworX 10.1.6, the **Global Script Namespace Declarations** field lists namespaces available to every C# and VB.NET script of the solution, with no per-script declaration: Script Tasks, Script Classes, expressions and display code-behind, on the server and on the clients.

- Separate the namespaces with semicolons, for example `System.Text.Json;MyCompany.Common`.
- Enter plain dotted namespace names. A pasted `using` or `Imports` prefix is removed. Aliases and `using static` are not supported, and invalid entries are ignored.
- A change applies at the next compile, with no Designer restart.
- The global namespaces also feed IntelliSense and appear in the predefined list of the per-script Namespace Declarations dialog.
- A namespace declaration does not add an assembly to the build. For a namespace from an external DLL, add the DLL in **Scripts References**. See Scripts References Reference.

In 10.1.5, namespaces are declared per script only, in the Namespace Declarations dialog of the Code Editor.

Provider Support

Database Providers

Provider	Databases Supported	Use Case
ODBC	Multiple via DSN	Legacy systems
OleDb	Multiple via connection	Windows databases
SQLite	SQLite files	Embedded/local
SQLClient	SQL Server	Enterprise
PostgreSQL	PostgreSQL	Open source
OracleClient	Oracle	Enterprise Oracle

Configuration Guidelines

Platform Selection

Windows Only - Choose when using:

- Windows Authentication
- OPC Classic
- Canary Historian
- Windows-specific .NET libraries

Multiplatform - Choose when:

- Deploying to Linux
- Using Docker containers
- Cross-platform requirement
- Cloud deployment

License Configuration

Must match production license:

If License Is	Set Family To	Set Model To
FrameworX-5000	FrameworX	5000
FactoryStudio-1500	FactoryStudio	1500
EdgeMachine-300	EdgeMachine	300

Culture Info Settings

Culture	Format Example	Use For
en-US	12/31/2024, $1,234.56	US English
pt-BR	31/12/2024, R$ 1.234,56	Brazilian Portuguese
de-DE	31.12.2024, 1.234,56 €	German
ja-JP	2024/12/31, ¥1,234	Japanese

Best Practices Checklist

Initial Setup

- Set Target Platform before development
- Configure Product Model to match license
- Set Culture Info for target locale
- Configure Python if using scripts

Retentive Database

- Use SQLite for standalone
- Use SQL Server for enterprise
- Test connection before saving
- Document connection string

Module Execution

- Script Tasks and Device Channel(s): leave both disabled unless the hardware is severely resource-constrained (limited CPU, limited memory, very few modules/channels). See *Run Modules Inside Main Process* above.
- Monitor Process Name in the Devices runtime monitor to confirm which execution mode is active.

Troubleshooting

Issue	Cause	Solution
Platform grayed out	Solution has objects	Remove objects first
License mismatch warning	Wrong model	Match to actual license
Python scripts fail	Wrong path	Verify interpreter location
Database connection fails	Wrong credentials	Test connection

In this section...
