Deployment Reference

Deploy FrameworX runtimes across servers, clients, and containers.

This section covers deploying FrameworX solutions to production environments, including runtime servers, client installations, and multi-tier architectures.

Overview
Deployment Architecture
Deployment Scenarios
Pre-Deployment Checklist
Quick Start by Scenario
Deployment Methods Comparison
TWebServices Role in Deployment
Child Pages
Next Steps

Overview

Solution deployment involves installing and configuring your developed solutions in production environments. This includes setting up runtime servers, configuring client access, establishing security, and ensuring high availability.

Unlike the simple Designer installation (1-3 minutes), production deployment requires:

- Firewall configuration
- License verification
- Gateway setup for multi-site
- Client display deployment
- Service configuration
- Security hardening

Deployment Architecture

Typical Production Setup

Runtime Server(s):

- TServer.exe (Runtime Engine)
- TWebServices (Remote Access)
- Database (SQLite/SQL Server)

Client Layer:

- Rich Clients (Windows)
- Web Clients (HTML5)
- Mobile Clients

Optional Components:

- TSecureGateway (Multi-site)
- Redundant Servers

Deployment Scenarios

Windows Runtime Deployment

For Windows-only solutions targeting .NET Framework 4.8:

- **Same installation as Designer** - Use full FrameworX platform installation
- **No separate runtime installer** - Configure installed platform for runtime-only use
- **Production setup** - Disable Designer access, configure auto-start, run as service

Multiplatform Runtime Deployment (cross-platform .NET)

For Linux, Docker, or edge devices, the runtime uses the cross-platform .NET Desktop Runtime: **.NET 8** for FrameworX 10.1.4 and earlier, **.NET 10** for FrameworX 10.1.5 and later. Three deployment options are available:

Option	Components	Size	Use Case
Runtime-Only	TServer, drivers, minimal files	~100 MB	Production edge devices
Full Platform	All files including Designer	~500 MB+	Enable remote ClickOnce Designer
TWebServices-Only	Communication server only	Minimal	Development/testing

The Full Platform option allows Windows computers to connect via web browser and launch Designer through ClickOnce with zero local installation.

Pre-Deployment Checklist

System Preparation

- Verify hardware meets production requirements
- Operating system updated and hardened
- .NET runtimes installed (for multiplatform: .NET 8 for 10.1.4 and earlier, .NET 10 for 10.1.5 and later)
- Firewall rules configured (ports 10108, solution-specific)
- Network connectivity verified
- Backup strategy defined

Solution Validation

- Solution tested in development environment
- Performance benchmarked with expected load
- Security review completed
- Documentation prepared for operators
- Rollback plan established

Licensing

- Runtime licenses acquired and activated
- License server configured (if using floating licenses)
- Client access licenses verified
- Gateway licenses ready (if multi-site deployment)

Quick Start by Scenario

Small Single-Server System

1. Install FrameworX on Windows server (full platform)
2. Configure TWebServices for remote access
3. Deploy solution files to server
4. Configure runtime as Windows Service
5. Connect clients to server

Enterprise Multi-Site

1. Deploy central server with full platform
2. Install TSecureGateway for site isolation
3. Configure gateway routing for each site
4. Deploy edge servers at remote locations
5. Establish secure network connections
6. Configure clients at each site

Linux/Edge Deployment

1. Choose deployment option (runtime-only vs full)
2. Install the matching cross-platform .NET runtime on the device (.NET 8 for 10.1.4 and earlier, .NET 10 for 10.1.5 and later), then install FrameworX with the TWebServices bootstrap and **Install Product** from Solution Center, as described in Linux Deployment Reference. Do not unzip or copy the Windows installation onto Linux: the folder layout differs.
3. Configure systemd service
4. Set up TWebServices if remote access needed
5. Deploy web clients for operator access

Container Deployment

1. Pull Docker image
2. Configure persistent volumes for data

3. Set up container
4. Upload solution
5. Run solution

Deployment Methods Comparison

Method	Complexity	Typical Size	Best For
Windows Full Platform	Simple	Complete installation	Windows-only environments
Linux Runtime-Only	Medium	< 5,000 tags	Edge devices, embedded systems
Full Platform on Linux	Medium	Any size	Multi-version environments
Docker Containers	Advanced	Scalable	Cloud, microservices
Redundant Servers	Complex	Mission-critical	High availability required

---

TWebServices Role in Deployment

TWebServices (port 10108) provides multiple functions:

- **Project Server**: Remote solution configuration
- **Client Gateway**: Web and mobile client connections
- **ClickOnce Host**: Designer deployment for Full Platform option
- **Solution Push**: Remote deployment in development (TWebServices-only option)

Configure based on deployment needs - full service for remote access or minimal for isolated systems.

Child Pages

- Runtime Installation (Reference) - Installing runtime components
- Client Types (Reference) - Deploying client displays
- Server Configuration (Reference) - Runtime server setup
- High Availability (Reference) - Redundancy and failover
- Container Deployment (Reference) - Deploying Docker application
- Security Hardening (Reference) - Production security
- Linux Deployment (Reference) - Deploying Linux application

Next Steps

After deployment:

- Monitor system performance
- Configure alarms and notifications
- Set up data archiving
- Establish maintenance procedures
- Train operators on system use

For basic installation: Installation & Licensing (Reference)
For development: Solutions (Guide)

In this section...
