Server Configuration Reference

Configure server instances and settings.

Server Configuration (Reference): Server Configuration (Reference): Configuring FrameworX runtime servers for production operations.

Server Architecture
Runtime Configuration
Solution Merging on Startup
Network Configuration
Performance Tuning
Monitoring
Backup and Recovery

Server Architecture

Core Components

Component	Function	Default Port
TServer.exe	Runtime engine	N/A (internal)
TWebServices	Remote access, web clients	10108
Database	Tag values, history, alarms	1433 (SQL Server)
TSecureGateway	Multi-site routing	Configurable

Runtime Configuration

Startup Settings

Configure in Runtime Startup:

Auto-Start Options:

- Start when computer starts
- Start as Windows Service
- Delay start (seconds)
- Auto-restart on failure

Command Line Parameters:

cmd

```
TServer.exe /solution:MySolution.tproj
  /port:10108
  /disableinterface
  /trace:verbose
```

Memory Management

Large Solutions (>50,000 tags):

xml

```
<runtime>
  <gcServer enabled="true"/>
  <gcConcurrent enabled="true"/>
  <GCHeapCount>8</GCHeapCount>
</runtime>
```

Solution Merging on Startup

The platform supports modular project design by automatically merging multiple project files at runtime based on naming conventions.

How It Works:

When launching a main solution (e.g., `MyProject`), FrameworX automatically merges any projects whose names:

- Start with the same base name
- Include a suffix with double underscores: `__suffix__`

Example Structure:

```
MyProject.tproj                      Main solution (core UI, shared logic)
MyProject__DeviceConfig1__.tproj     Device-specific configurations
MyProject__DBTest__.tproj            Test database connections
MyProject__CustomerA__.tproj         Customer-specific customizations
```

Runtime Behavior:

When you start `MyProject`, the runtime automatically:

1. Detects all matching `MyProject__*__.tproj` files
2. Merges their contents (tags, scripts, displays, connections)
3. Creates a unified runtime environment
4. Preserves modular structure for development

Benefits:

- **Modular Development**: Teams can work on separate modules
- **Environment-Specific**: Different configs for dev/test/production
- **Customer Variants**: Maintain base product with customizations
- **Version Control**: Easier to manage smaller, focused files

Merge Priority:

If conflicts exist (same tag name in multiple files):

- Main project takes precedence
- Modules merged in alphabetical order
- Last definition wins for duplicates

Configuration Example:

```
Solutions/
??? MyProject.tproj
??? MyProject__Common__.tproj
??? MyProject__PlantA__.tproj
??? MyProject__PlantB__.tproj
??? MyProject__Debug__.tproj
```

Starting `MyProject` loads all modules automatically.

Network Configuration

Firewall Rules

Required Ports:

Port	Protocol	Direction	Purpose
10108	TCP	Inbound	TWebServices/Clients
1433	TCP	Outbound	SQL Server
502	TCP	Outbound	Modbus devices
102	TCP	Outbound	S7 PLCs

Windows Firewall:

cmd

```
netsh advfirewall firewall add rule name="FrameworX Runtime"
  dir=in action=allow protocol=TCP localport=10108
```

Network Adapters

For multi-homed servers:

xml

```xml
<networkSettings>
  <bindToIP>192.168.1.100</bindToIP>
  <clientInterface>192.168.1.0/24</clientInterface>
  <deviceInterface>10.0.0.0/24</deviceInterface>
</networkSettings>
```

Performance Tuning

Windows Optimization

Power Settings:

cmd

```
powercfg /setactive 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c
```

(High Performance plan)

Process Priority:

cmd

```
wmic process where name="TServer.exe" CALL setpriority "high"
```

Linux Optimization

System Limits:

bash

```
# /etc/security/limits.conf
frameworx soft nofile 65536
frameworx hard nofile 65536
frameworx soft nproc 32768
frameworx hard nproc 32768
```

CPU Affinity:

bash

```
taskset -c 0-3 /opt/frameworx/TServer.exe
```

Monitoring

Performance Counters

Monitor these metrics:

- CPU usage < 70%
- Memory usage < 80%
- Disk queue length < 2
- Network utilization < 60%

Log Configuration

xml

```xml
<logging>
  <level>Information</level>
  <maxFileSize>100MB</maxFileSize>
  <maxFiles>10</maxFiles>
  <path>C:\Logs\FrameworX\</path>
</logging>
```

Health Checks

Configure endpoint for monitoring:

```
http://server:10108/health
```

Returns: CPU, Memory, Disk, Service Status

Backup and Recovery

Backup Strategy

Daily Backup:

1. Stop runtime (if possible)
2. Backup solution files (.tproj, .dbsln)
3. Backup database
4. Backup configuration files
5. Restart runtime

Backup Script:

powershell

```
Stop-Service "FrameworX Runtime"
Copy-Item "C:\Solutions\*" "\\backup\solutions\" -Recurse
Start-Service "FrameworX Runtime"
```

Disaster Recovery

RPO/RTO Targets:

- Recovery Point Objective: 1 hour
- Recovery Time Objective: 15 minutes

Recovery Steps:

1. Install FrameworX on new server
2. Restore solution files
3. Restore database
4. Update client connections
5. Verify operation

In this section...
