Runtime Engine Reference

Understand the runtime engine architecture.

Reference Solution Runtime UI | Engine | Troubleshooting | Diagnostics | Client-Server

---

The Runtime Engine (Reference): The core execution component managing real-time data processing, module coordination, and client services in production environments.

Core Process - TServer.exe
Memory Architecture
Process & Thread Model
Data Flow Architecture
Module Execution
Performance Characteristics
Optimization Settings
Diagnostics & Monitoring
High Availability

Core Process - TServer.exe

TServer.exe is the computer process maintaining the real-time database. During execution, it:

- Loads Tags, Templates, and Assets into memory
- Acts as central reference point for all modules
- Manages microsecond-level updates
- Can run as Windows Service or on Linux/other OS

TServer Components

Component	Function	Key Features
Core Engine	Central coordinator	Module management, event routing, memory management
Tag Services	Real-time database	In-memory storage, microsecond updates, quality management
Module Host	Module execution	Dynamic loading, lifecycle management, isolation
Communication Layer	Network services	Client connections, protocol handlers, buffering
Security Context	Access control	Authentication, authorization, audit logging

---

Memory Architecture

Runtime Memory Space

Memory Structure:

- **Real-time Tag Database (In-Memory)**
  - Current Values
  - Quality & Timestamps
  - Cached Metadata
- **Module Memory**
  - Device Buffers
  - Alarm States
  - Script Context
  - Client Sessions
- **System Cache**
  - Configuration
  - Security Tokens
  - Performance Metrics

Memory Management

Area	Purpose	Management
Tag Database	Current values storage	Automatic optimization
Module Memory	Processing buffers	Per-module allocation

System Cache	Performance data	LRU cache policy

Process & Thread Model

TServer Main Process

Modules - Services Thread Pool:

- TagProvider services
- Alarm processing
- Historian services
- Datasets & SQL services
- Solution DB & cached metadata

Client Services Threads:

- TCP Listener Thread
- Client Handler Threads (1 per client)
- Web Service Thread Pool

System Threads:

- Garbage Collection
- Performance Monitoring
- Diagnostics Logging

Script Module Process

Script Task Execution:

Mode	Threading	Configuration
Default	One thread per task	Standard isolation
Multi-thread	Multiple threads	Parallel execution
Configurable	Max threads setting	Resource control

Script Expression Execution:

- Single thread, event-driven execution
- Optimized for rapid response

Script Classes:

- Client service threads
- Instance per session

Data Flow Architecture

Real-Time Data Flow

Flow Sequence:

```
Field Devices  Device Module  Tag Database  Processing Modules  Clients
                                  Historian
                                  Database
```

Data Flow Stages:

Stage	Processing	Latency
Acquisition	Device polling/events	1-100ms
Tag Update	Memory write	<1ms
Processing	Alarms/Scripts	1-10ms

Distribution	Client notification	10-50ms
Storage	Historian/Database	50-500ms

Event-Driven Architecture

Event Processing:

Event Type	Priority	Handler
Tag Change	High	Direct notification
Alarm	Critical	Immediate processing
Script Trigger	Normal	Queued execution
Client Request	Variable	Thread pool
System	Low	Background thread

Module Execution

Module Lifecycle

Phase	Actions	Duration
Initialize	Load configuration	100-500ms
Start	Begin processing	10-100ms
Run	Active execution	Continuous
Stop	Graceful shutdown	100-1000ms
Dispose	Release resources	10-100ms

Module Isolation

Each module runs with:

- Dedicated memory space
- Independent error handling
- Configurable resource limits
- Separate logging context

Performance Characteristics

Processing Capacity

Metric	Typical	Maximum
Tag Updates/sec	10,000	100,000+
Concurrent Clients	50	500+
Script Executions/sec	1,000	10,000+
Alarm Processing/sec	5,000	50,000+

Resource Usage

Component	CPU Usage	Memory
Core Engine	5-10%	100-200 MB
Tag Services	10-20%	Varies with count

Modules	20-40%	50-500 MB each
Client Services	5-10% per client	10-50 MB per client

Optimization Settings

Thread Pool Configuration

Setting	Default	Range	Impact
Min Threads	4	2-16	Startup performance
Max Threads	100	10-1000	Peak capacity
Queue Size	1000	100-10000	Buffer capacity

Memory Settings

Parameter	Default	Recommended
Initial Heap	256 MB	512 MB for large systems
Max Heap	2 GB	4-8 GB for enterprise
Tag Cache	100 MB	Scale with tag count

Diagnostics & Monitoring

Performance Counters

Monitor these metrics:

- CPU utilization by module
- Memory consumption trends
- Thread pool saturation
- Queue depths
- Response times

Log Categories

Category	Content	Verbosity
System	Core operations	Error/Warning
Modules	Module events	Info
Communication	Client traffic	Debug
Performance	Metrics	Trace

High Availability

Redundancy Support

Feature	Implementation	Switchover Time
Hot Standby	Synchronized backup	Activation under 1 second; typically under 5 seconds end to end
Data Replication	Real-time sync	Continuous
Client Redirect	Automatic failover	2-5 seconds

A switchover has three phases: detection, activation and client redirect. See the Failover sequence and timing section of Redundancy Reference.

State Synchronization

Between primary and backup:

- Tag values and quality
- Alarm states
- Client sessions
- Module status

In this section...
