Historian Tags Reference

Define tags to be historized.

Reference Modules Historian UI Tags | Tables | Storage | Monitor

---

Historian Tags (Reference) define the mapping between data points and their archival storage configuration within the FrameworX Historian module. Historian Tags are the fundamental configuration objects that:

- Link tags to historian tables for organized storage
- Define data collection parameters
- Configure deadband and deviation settings
- Support both internal tags and external assets
- Enable template-based archiving

Each Historian Tag represents a data point tracked and stored over time with specific collection rules.

Configuration Properties
Creating Historian Tags
Deadband Configuration
DeviationDeadBandType
UserType Tags (Template Tags)
HistorianTable Assignment
Runtime Behavior
Performance Optimization
Common Configurations
Troubleshooting
Best Practices Checklist

Configuration Properties

Property	Description	Required	Default
TagName	Unique identifier of tag to archive	Yes	-
HistorianTable	Target table for data storage	Yes	-
DeadBand	Minimum change in units for SaveOnChange	No	0
Deviation	Override for TimeDeadBand interval	No	0
RateOfChange	Units/second threshold for storage	No	0
DeviationDeadBandType	Absolute or Percentual calculation	No	Absolute

---

Creating Historian Tags

Prerequisites

- Configured HistorianTable (or use default Table1)
- Tags defined in Unified Namespace
- Understanding of collection requirements

Configuration Steps

1. Navigate to **Historian Historian Tags**
2. Enter **TagName** or select from list
3. Configure deadband parameters:
    - **DeadBand** for value changes
    - **Deviation** for time overrides
    - **RateOfChange** for velocity triggers
4. Select **DeviationDeadBandType**
5. Choose target **HistorianTable**

Deadband Configuration

DeadBand (Value Change)

Minimum absolute change required for storage:

- Applied with SaveOnChange trigger
- Reduces database writes
- Filters noise and minor fluctuations

Example:

```
TagName: Tank.Level
DeadBand: 0.5
// Only stores when level changes by ±0.5 units
```

Deviation (Time Override)

Overrides TimeDeadBand interval when exceeded:

- Forces storage regardless of time interval
- Captures significant changes immediately
- Works with TimeDeadBand configuration

Example:

```
TagName: Reactor.Temperature
Deviation: 5.0
TimeDeadBand: 60 seconds
// Stores every 60s OR when temp changes by ±5°
```

RateOfChange (Velocity)

Triggers on rapid value changes:

- Measured in units per second
- Detects process upsets
- Overrides time intervals

Example:

```
TagName: Pressure.Line1
RateOfChange: 10
// Stores when pressure changes >10 PSI/second
```

DeviationDeadBandType

Absolute

Fixed unit threshold:

```
Current Value: 100
Deviation: 5
Triggers at: <95 or >105
```

Percentual

Percentage of current value:

```
Current Value: 100
Deviation: 5 (%)
Triggers at: <95 or >105

Current Value: 200
Deviation: 5 (%)
Triggers at: <190 or >210
```

UserType Tags (Template Tags)

Archive entire templates or specific properties:

Full Template

```
TagName: Pump1
// Archives all properties of Pump1 template
```

Specific Property

```
TagName: Pump1.Motor.Current
// Archives only motor current property
```

Multiple Instances

When using template arrays:

```
TagName: Pumps[1].Flow
TagName: Pumps[2].Flow
// Individual configuration per instance
```

HistorianTable Assignment

Grouping Strategy

Assign tags to tables based on:

- Update frequency
- Process relationships
- Data types
- Storage requirements

Best Practices

Update Rate	Table Assignment
< 1 second	HighSpeed
1-10 seconds	Process
10-60 seconds	Supervisory
> 60 seconds	SlowData

Runtime Behavior

SaveOnChange Mode

```
Tag changes  Check DeadBand  Store if exceeded
```

Trigger Mode

```
Trigger fires  Store all table tags  Apply timestamp
```

TimeDeadBand Mode

```
Timer expires  Check Deviation/RateOfChange  Store if exceeded OR timer
```

Performance Optimization

DeadBand Settings

- Analog signals: 0.1-1% of range
- Digital signals: Usually 0 (all changes)
- Calculated values: Based on significance

Table Organization

- Group similar update rates
- Separate critical from non-critical
- Consider query patterns
- Balance table sizes

Storage Efficiency

- Use Normalized tables for sparse data
- Standard tables for synchronized data
- Enable compression where supported

Common Configurations

Temperature Monitoring

```
TagName: Furnace.Temperature
DeadBand: 0.5
Deviation: 10
RateOfChange: 5
DeviationDeadBandType: Absolute
HistorianTable: ProcessData
```

Level Tracking

```
TagName: Tank.Level
DeadBand: 0.1
Deviation: 5
DeviationDeadBandType: Percentual
HistorianTable: TankFarm
```

Discrete Events

```
TagName: Pump.Status
DeadBand: 0
HistorianTable: Events
// Captures all state changes
```

Troubleshooting

Data not archiving:

- Verify tag name spelling
- Check HistorianTable is active
- Review deadband settings
- Confirm tag has valid quality

Excessive storage:

- Increase deadband values
- Review deviation settings
- Check for oscillating tags
- Enable time deadband

Missing rapid changes:

- Decrease RateOfChange threshold
- Reduce deviation value
- Check scan rates
- Verify trigger configuration

Template tags not working:

- Confirm template structure
- Check property paths
- Verify array indices
- Review instance names

Best Practices Checklist

- [ ] **Set appropriate deadbands** - Balance detail vs storage
- [ ] **Group related tags** - Same table for correlated data
- [ ] **Document settings** - Record why values chosen
- [ ] **Test configurations** - Verify before production
- [ ] **Monitor performance** - Track storage growth
- [ ] **Use templates efficiently** - Archive needed properties only
- [ ] **Plan for maintenance** - Consider future adjustments

In this section...