

SQL Diagnostic Manager

Getting Started Guide

From purchase to productive — faster than you think.



Welcome — You Made a Smart Investment


SQL Diagnostic Manager is one of the most capable SQL Server performance monitoring platforms available. You've made an excellent choice — and this guide is going to make sure you feel that from day one.

 Trusted by thousands of DBAs worldwide to keep mission-critical SQL Server environments healthy.

 Works for teams of every size — solo DBA or enterprise shop — from on-premises to cloud to hybrid.

 Agentless architecture means nothing gets installed on your monitored SQL Server instances.

 You don't have to figure this out alone — Idera's documentation, knowledge base, and support team have your back.

 Most teams have meaningful monitoring live within 30 minutes of install. Let's get you there today.

What Is SQL Diagnostic Manager?

SQL Diagnostic Manager (SQLdm) is an agentless SQL Server performance monitoring platform. It watches your instances around the clock — on-premises, cloud, or hybrid — giving you visibility into what's happening and the tools to act on it.



Real-Time Monitoring

Live dashboards for CPU, memory, I/O, disk, TempDB, and network across every registered instance.



Intelligent Alerting

Threshold-based and baseline-relative alerts delivered via email, SNMP, or automated response scripts.



Root Cause Diagnostics

Navigate from an active alert directly to blocking chains, wait stats, or query details in seconds.



Historical Analysis

Long-term repository for forensic investigation of past incidents and capacity planning.



Query Analysis

Capture and analyze slow, blocking, and deadlocking queries — without installing anything on target servers.



Reporting

Built-in SQL Reporting Services integration for scheduled reports on availability, performance, and trends.



Baseline & Predictive Analysis

Learns what 'normal' looks like per instance and forecasts future alert conditions using historical patterns.



Alert Automation

Configure Alert Actions & Responses to auto-execute scripts, kill blocking sessions, or notify external systems.

Core Concepts — Terms You'll See Throughout This Guide

New to SQLdm? Bookmark this slide. These are the terms you'll encounter most.

Snapshot

A point-in-time collection of performance data from a monitored SQL instance. SQLdm takes one every 6 minutes by default.

Collection Service

The SQLdm component that polls your monitored instances via DMVs. Runs on Windows — nothing installed on target servers.

DMV

Dynamic Management View — built-in SQL Server system views that expose performance data. SQLdm queries these to collect metrics.

Wait Stats

SQL Server's record of what queries are waiting on — locks, I/O, memory, etc. One of the most powerful root-cause tools available.

Repository

The SQL Server database where SQLdm stores all collected metrics, alerts, configuration, and history.

Baseline

A learned picture of normal performance for an instance, built from the last 7 days of data by default. Windows are customizable.

Query Monitor

Optional feature that captures slow, blocking, or deadlocking queries per instance. Disabled by default; configured per instance.

Alert Threshold

The value at which SQLdm fires a warning (yellow) or critical (red) alert. Every threshold is customizable per instance or template.

How to Use This Guide

This guide walks you through setup and configuration in a logical order — from install to your first active alerts to deeper tuning. Each section builds on the last.

1



Install & Connect

System requirements, service accounts, ports, and adding your first SQL instances.

2



Your First Win

Get live monitoring running and your first real alert configured in 30 minutes.

3



Core Configuration

Alert thresholds, baselines, Query Monitor, Server Tags, and templates.

4



Day 1 / Week 1 / Week 2

A practical timeline to get from installed to fully tuned without overcomplicating it.

5



Use Cases & Resources

Real scenarios SQLdm solves every day, plus documentation and support links.

SECTION 1

Install & Connect

Everything you need before adding your first instance



Installation Checklist

Run through this list before adding your first SQL instance — it takes 10 minutes and sets you up for a smooth start.



SQL Server 2016+ for the Repository database

The SQLdm Repository requires SQL Server 2016 or later. Your monitored SQL Server instances can be any supported version.



Windows Server with .NET Framework

All SQLdm services (Management and Collection) run on Windows. The installer checks for .NET and prompts if an update is needed.



Service account with sysadmin on monitored instances

The Collection Service needs sysadmin access to poll DMVs. No agent is deployed to target servers.



Ports 5166 (Management) and 5167 (Collection) open

Required for inter-service communication. Port 1433 (or your configured SQL port) must be reachable from the Repository.

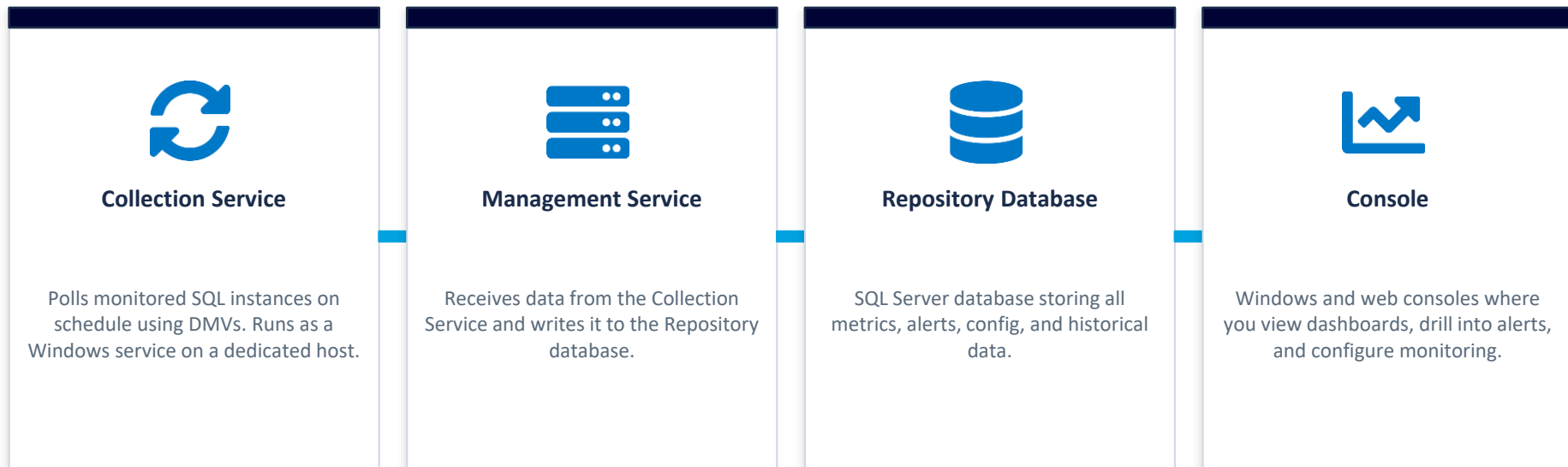


Repository autogrow configured before production

Default autogrow settings are fine for evaluation. Increase file growth increments before adding more than a handful of monitored instances.

How SQLdm Works — Architecture Overview

SQLdm uses a four-component architecture. Nothing is installed on your monitored SQL Server instances — monitoring is entirely agentless.



✓ Agentless — nothing on monitored servers ✓ Scales to 300+ instances per Collection Service¹ ✓ On-premises, cloud, or hybrid

¹ Additional Collection Services can be deployed as needed — SQLdm's scale-out architecture supports up to 2,000 monitored instances.

Adding Instances & Server Tags

Start with your three most critical SQL instances. Server Tags let you group and manage them from the moment you add them.

Adding a SQL Instance

- 1 Right-click in the console navigation pane → Add SQL Server
- 2 Enter the server name or IP and choose your authentication method (Windows auth recommended)
- 3 Accept default monitoring settings — you can tune these per instance after the first snapshot
- 4 SQLdm begins collecting within seconds; your first full snapshot arrives within 6 minutes

Server Tags — Why They Matter



Group instances logically

Tag by environment (Prod/Dev/QA), application, geography, or team — any label that makes sense for your org.



Filter dashboards and alerts

View only the instances relevant to your current focus — a single app, a single region, or a single tier.



Apply settings in bulk

Push alert threshold templates or monitoring settings to all instances sharing a tag in one action.



Scope access by team

Combine with security roles to give each team visibility into only their tagged instances.

SECTION 2

Your First Win

Live monitoring and your first real alert in 30 minutes



Your First Win — Live Monitoring in 30 Minutes

Within 30 minutes of adding your first instance you can have meaningful monitoring in place. Here's the fastest path:

1

Add your most critical SQL instance

Right-click in the console → Add SQL Server. Provide the server name and credentials, then accept defaults.

2

Allow the first snapshot to complete

SQLdm collects a snapshot every 6 minutes by default. The first snapshot populates your dashboards with live data.

3

Open Active Alerts and review

Click the Active Alerts tab. Red = critical, yellow = warning. Review each one before making any changes — understanding what's firing is more important than silencing it.

4

Adjust one threshold that doesn't fit

Find an alert that's firing but doesn't reflect a real concern for this server. Right-click → Configure Alert Thresholds and tune it.

5

Set up email notifications

Tools → Alert Actions and Responses → configure SMTP. Set at minimum one email rule for Critical (red) alerts.

Your Goal

One instance live.
Alerts configured.
Email notification set.

SECTION 3

Core Configuration

Alerts, baselines, Query Monitor, views, and more



Alert Thresholds — Setting the Right Expectations

Default alert thresholds are starting points — they're built from general best practices and may not reflect the performance characteristics of your specific servers. Plan to tune them.

Tuning Approach



Expect some noise at first

Newly added instances will often fire several alerts. This is normal — it means monitoring is working. Investigate each one before adjusting.



Use baselines to tune smarter

After 7+ days of baseline data, SQLdm tells you what's normal for each instance. Use that data to set thresholds that actually mean something.



Use templates for consistency

Alert threshold templates let you define a standard configuration and push it to groups of instances — ideal when managing multiple servers with similar roles.

Common Alerts to Tune Early

OS Memory Usage

Review if SQL Server owns most available RAM by design

Session CPU Time

SPIDs reuse — cumulative values mean this needs a higher threshold

Fragmentation

Only clears after the next scheduled Table Statistics collection

Blocking Session

Wire to an Alert Action that auto-kills the blocking session

TempDB Usage

Enable baseline first — what's normal here varies widely

Page Life Expectancy

Learn your PLE baseline before setting a fixed threshold

Baseline Configuration — Teaching SQLdm What 'Normal' Looks Like

Baselining is one of SQLdm's most powerful capabilities. Once established, it lets you move from fixed alert thresholds to alerts that are relative to each server's actual behavior.



Default: last 7 days of collected data

SQLdm builds the baseline dynamically using the last 7 days of snapshots by default. No manual setup required — it starts learning from the moment you add an instance.



Customize the window to fit your environment

Right-click instance → Properties → Baseline Configuration. You can set a custom date range and restrict which days and hours are included (e.g., Mon–Fri, 8 AM–5 PM).



Baseline-relative alerts are smarter alerts

Once baseline is established, you can configure alerts to fire when a metric deviates from the learned norm rather than a fixed number — far more accurate for busy environments.



Apply the same baseline config to multiple servers

After configuring baseline settings on one instance, use the 'Apply Baseline Configuration to other SQL Servers' option to push it to your other instances.

Query Monitor — Capturing What's Slow

Query Monitor captures slow, blocking, and deadlocking queries without installing anything on your monitored servers. It's disabled by default and configured per instance.

Setting	Default	Recommended Starting Point	Where to Change
Query Monitor	Disabled	Enable per instance when ready to capture query data	Instance Properties → Query Monitor
Duration Threshold	5,000 ms	Start at 5,000ms — increase if you have known long-running queries or jobs	Instance Properties → Query Monitor
SQL Batch Capture	Off	Enable for full batch text alongside query stats	Instance Properties → Query Monitor

Things to Know About Query Monitor

- Duration threshold is per-query — if your environment has many short but high-frequency queries, use Wait Stats (see below) instead of lowering this threshold
- Query Monitor data grows your Repository — review grooming settings (Tools → Grooming) before enabling on many instances
- Results appear in the Query Monitor tab for each instance — sort by duration, CPU, or I/O to prioritize what to fix first

Wait Type Views — Understanding Query Waits

Wait Stats reveal what SQL Server is actually waiting on when queries are slow — locks, I/O, memory, CPU scheduling, and more.



Query Waits by Duration

Shows the total elapsed wait time per query — best for identifying which individual queries are spending the most time waiting. Use this when a specific query is slow.

When: a user-reported slow query, a flagged item from Query Monitor, or a high-duration alert.



Query Waits by Time

Shows aggregate wait time across a collection window — best for identifying systemic resource pressure. Use this when the whole system feels slow rather than one specific query.

When: broad slowdowns, high CPU or I/O alerts, or preparing for a capacity conversation.

Alert Actions & Responses — Make SQLdm Do the Work

Alert Actions and Responses let SQLdm act automatically when an alert fires — not just notify you, but actually respond. This is one of the most underused and highest-value features in the product.



Email Notification

Send an email to one or more addresses when a specific alert condition is met. Filter by severity, instance, or alert type.



Execute a SQL Script

Run a T-SQL script when an alert fires — ideal for auto-killing a blocking session, capturing a diagnostic snapshot, or logging to a custom table.



Run a Windows Program

Launch an executable or script on the SQLdm server when an alert fires — useful for triggering external notification systems.



Log to Windows Event Log

Write alert events to the Windows Application Event Log for integration with monitoring platforms that watch event logs.



SNMP Trap

Send an SNMP trap to a network management system — integrates SQLdm alerts with enterprise monitoring infrastructure.



Scope by Alert Type

Configure different responses for different alerts — aggressive auto-response for blocking sessions, email-only for disk warnings.

Configure under: Tools → Alert Actions and Responses

Administration, Security & Custom Counters

Security & Access Control



Role-based access

Assign users to SQLdm roles (Administrator, User, Guest) to control who can configure monitoring vs. who can only view it.



Scope by Server Tag

Combine roles with Server Tags to give team members access to only the instances relevant to their responsibilities.



Windows Authentication

SQLdm uses Windows auth for console login — no separate credential store. Active Directory groups are supported.

Custom Counters

Custom Counters let you monitor any Windows Performance Monitor metric alongside SQLdm's built-in metrics — application-level counters, IIS metrics, custom application KPIs, or any PerfMon-exposed value.



Any PerfMon counter

If Windows Performance Monitor can see it, SQLdm can track it — IIS request rates, .NET CLR metrics, custom app counters.



Alertable

Custom Counters can be configured with warning and critical thresholds just like built-in metrics.



Trend over time

Historical counter data is stored in the Repository — useful for correlation and capacity planning.

Configure under: Tools → Custom Counters

SECTION 4

Day 1 · Week 1 · Week 2

A practical timeline — not a project plan



Your Getting Started Timeline

SQLdm is fast to set up. This timeline gets you from installed to fully tuned — without making it more complicated than it needs to be.

Day 1

Get monitoring live

- ✓ Install SQLdm and complete the checklist
- ✓ Add your 3 most critical SQL instances
- ✓ Configure Server Tags for your instances
- ✓ Verify first snapshot runs without errors
- ✓ Set up email alerts for Critical (red) alerts

Week 1

Configure and learn

- ✓ Review Active Alerts — investigate before tuning
- ✓ Enable Query Monitor on your busiest instance (start at 5,000ms)
- ✓ Enable baseline collection — it needs 7 days to mature
- ✓ Explore Wait Type views during your next peak period
- ✓ Configure Alert Actions & Responses for Critical alerts

Week 2

Tune and refine

- ✓ Review baseline data — adjust alert thresholds to match observed normal
- ✓ Review Query Monitor results — tune duration threshold based on what you captured
- ✓ Configure Alert Actions & Responses (auto-kill script for blocking sessions)
- ✓ Add remaining SQL instances with Server Tags applied
- ✓ Set up Custom Counters for any app-level metrics you track

SECTION 5

Real Use Cases

What teams actually use SQL Diagnostic Manager for every day



Real Use Cases — SQL Diagnostic Manager in Practice



"Users say the app is slow"

Open Active Alerts → sort by severity. Check Blocking Sessions and Query Monitor top queries. You'll have a root cause in minutes, not hours.



"We need to prove SLA compliance"

Use the History Browser to pull CPU, wait stats, and availability for any date range. Export to SQL Reporting Services for executive-ready reports.



"Disk space is running out"

SQLdm tracks disk capacity trends. Set a disk space alert before you hit 90% and get notified before the 11 PM call comes in.



"Something changed last night — what was it?"

Historical Snapshots let you compare performance across any two time windows. Combine with baseline deviation alerts to pinpoint when behavior changed.



"We have a blocking problem"

Configure an Alert Action on Blocking Session alerts to auto-kill the head blocker and capture a diagnostic T-SQL snapshot for post-mortem review.



"Management wants a performance report"

Use the built-in SSRS reports for trend analysis, availability summaries, and top query consumers. Schedule weekly delivery to stakeholders.

Common Mistakes — and How to Avoid Them


Setting Query Monitor duration to zero or very low right away

 Start at 5,000ms — the default. Lower it gradually based on what you actually see captured. Too low means noise and Repository growth.

Leaving all alert thresholds at defaults and never tuning them

 Default thresholds don't know your servers. Give baselines 7 days to mature, then tune thresholds to match what's actually normal for each instance.

Assuming fragmentation alerts clear after a manual index rebuild

 Fragmentation alerts only clear after the next scheduled Table Statistics collection runs. After a rebuild, either reschedule collection or manually clear the alert.

Ignoring data retention until the Repository gets large

 Query Monitor and Wait Stats data grows the Repository. Review grooming settings (Tools → Grooming) before enabling on many instances.

Thank You for Choosing SQL Diagnostic Manager.

We're genuinely glad you're here. SQL Diagnostic Manager is a powerful product — and you made a smart investment. The team at Idera is committed to making sure you feel that from day one and every day after.

Your next step is simple:

Open SQL Diagnostic Manager, add your most critical SQL instance, and let it run its first snapshot. Everything else in this guide follows naturally from there.

Documentation

wiki.idera.com/display/SQLDM

What's New

wiki.idera.com/display/SQLDM/New+features

Support Portal

idera.com/support/productsupport

Product Site

idera.com/products/sql-diagnostic-manager

IDERA

Idera · Helping teams build, manage, and protect data for over 25 years.