Service Health Monitor – Setup & User Guide

1. 📘 Introduction

The **Service Health Monitor Tool** is designed to provide real-time visibility and status tracking for all **Contentverse Windows services** and **Tomcat-based applications**. It offers a unified interface for monitoring system health, performance metrics, and service availability—helping teams maintain uptime, diagnose issues faster, and ensure smooth application performance.

🔍 **Key Features:**

- **Windows Services Monitoring**
  - Real-time status (Running/Stopped)
  - CPU usage
  - Memory consumption
  - Active connections
  - Visual graphs for CPU and memory trends
  - Access to service-specific logs
- **Tomcat Server Monitoring**
  - Thread activity tracking
  - JVM memory usage
  - Application health status
  - API access logs & STD error logs
  - Server overview panel
- **Smart Notifications**
  - Instant **in-app alerts**, **email notifications**, and **SMS messages** for service up/down events

With its intuitive dashboard and powerful monitoring capabilities, this tool is essential for any Contentverse deployment where service reliability and responsiveness are critical.

2. 🧰 Tomcat Server – Monitoring Panel

📄 Overview

The **Tomcat Server** appears by default on the left-hand sidebar under the "Services" section. It is a non-removable core monitoring unit but **can be configured or edited** if needed. *(Refer to: “How to Edit a Service” for configuration steps.)*

The central panel is split into:

- A **navigation panel** (left) with different monitoring tabs
- A **data display section** (right) showing real-time metrics
- A **"Last Updated" timestamp** (top-right), which refreshes every 5 seconds

---

1️⃣Server Overview

Located under **Tomcat Manager > Server Overview**, this tab provides essential runtime and configuration data of the Tomcat instance.

Metrics shown:

- ✅ **Server Status** – Indicates whether the Tomcat service is Running or Stopped.
- 🕒 **Uptime** – Shows the total time since the server was last started, along with the exact start timestamp.
- ⚙️ **JVM Version** – Displays the Java runtime in use (e.g., OpenJDK 1.8.0_442).
- 🖥️ **System Information** – Key OS and environment data:
    - **Operating System:** e.g., Windows_NT 10.0.26100
    - **Java Runtime:** e.g., OpenJDK 1.8.0_442
    - **Server Version:** e.g., Apache Tomcat/9.0.97

2️⃣Thread & Connection

This section gives insight into request handling and thread utilization.

Metrics shown:

- **Max Threads** – Maximum number of threads allocated (e.g., 200) and current active threads (e.g., 10).
- **Busy Threads** – Threads currently handling requests and % utilization.
- **Request Count** – Total requests served (e.g., 2840), with real-time error count.
- **Avg Processing Time** – Average response time in milliseconds (e.g., 3ms) and timeout setting.
- 📊 **Graphs:**
  - *Thread Usage Over Time:* Real-time chart of max vs. busy threads.
  - *Requests & Errors:* Tracks incoming requests and HTTP error frequency.

3️⃣ JVM & Memory

This tab visualizes Java memory allocation and usage.

Metrics shown:

- **Heap Memory:**
  - Used: e.g., 113 MB
  - Max: e.g., 246 MB (46.1%)
- **Non-Heap Memory:**
  - Used: e.g., 117 MB
  - Max: e.g., 1264 MB (9.2%)
- 📊 **Graphs:**
  - *Heap Memory Usage:* Comparison between max heap and used heap.
  - *Non-Heap Memory Usage:* Same comparison for non-heap memory.

4️⃣ Applications

This section lists all active web applications deployed on the Tomcat server.

Table includes:

- **Application Name** – e.g., ROOT, Repositories, CVPreferences
- **Context Path** – URL path (e.g., /, /CVPreferences)
- **Status** – Each application's running state
- **Sessions** – Number of active user sessions per app (often 0 unless used live)

🔄 This section is auto-refreshed every 5 seconds and is ideal for checking deployment status in real time.

5️⃣ Log Viewer

The final tab under **Tomcat Manager** provides live access to server logs.

Tabs:

- **API Access** – Shows real-time API requests, method type, endpoint path, and response codes
- **Std Error** – Displays stack traces or error messages if the application encounters runtime issues

This log viewer helps developers troubleshoot issues and analyze endpoint behavior without needing remote server access.

3. 🖥 Windows Services – Real-Time Monitoring

The **Windows Services** section is designed to monitor essential **Contentverse services** such as CV PDF, PDFViewerWebService, etc. These services need to be **manually registered** before they appear in the monitoring interface.
➡️ *(See: "How to Register a Service" for step-by-step instructions.)*

📌 Sidebar Status Indicators

Once registered:

- Each service appears under **"Windows Services"** in the left-hand **sidebar**
- A **green dot** indicates the service is currently **Running**
- A **red dot** indicates the service is **Stopped or Inactive**

📊 Metrics Tab

When a service is selected (e.g., CV PDF), the right-hand panel displays real-time health data:

Metrics shown:

- **Status** – Shows if the service is running
- **Memory Usage** – Percentage of RAM currently used by the service
- **CPU Usage** – Percentage of CPU consumed
- **Connections** – Active connection count (if applicable)

📈 Below these KPIs, a **Resource Usage Trend Graph** visualizes:

- **CPU** (blue line)
- **Memory** (purple line)
  ...over time, updating every 5 seconds for live tracking.

🗀 Logs Tab

Selecting the **Logs** tab provides complete service logs for the selected Windows service.

Features:

- Each log entry shows:
  - Log level (INFO, ERROR, WARNING)
  - Timestamp
  - Service name
- You can **search logs** using the search box
- Filter logs using the **"All Levels" dropdown**, allowing:
  - INFO
  - WARNING
  - ERROR
- Click **Export CSV** to download the current logs for offline analysis or support sharing.

📝 Logs update automatically and display the most recent entries at the top.

4. 🛠 How to Register a Windows Service

To start monitoring a Windows service in the **Service Health Monitor**, you must first register the service manually. This process ensures the system can track its metrics, logs, and status.

🔍 Steps to Register a Service

1. **Locate the ➕ icon**
    - Go to the **left sidebar** under **“Windows Services”**
    - Click the **+ icon** next to it
2. **Fill in the Registration Form**
   A pop-up titled **"Register Service"** will appear with the following fields:

✅ **Service Name**

- This must be **exactly the same** as the service name shown in your Windows system.
- 👉 To find it:
    - Press Win + R → type services.msc → press Enter
    - Find the Windows service you want to monitor (e.g., PDFViewerWebService)
    - Right-click it → select Properties
    - Copy the **Service Name** (not Display Name) and paste it into this field

🌐 **Service URL**

- This should be the **host URL** where the service is running
- Example: http://localhost

🔢 **Port Number**

- The port on which the service is hosted
- Example: 8005

3. **Click “Register”**
   Once the form is filled out, click the blue **“Register”** button.

🧪 Example Entry

Field	**Value**
Service Name	CV PDF
Service URL	http://localhost
Port Number	8005

After registration, the service will appear in the **sidebar** under Windows Services, and its metrics and logs will begin to update in real time.

5. ✏️ How to Edit a Service

Once a service is registered—whether it's a **Windows Service** or the built-in **Tomcat Server**—you may need to **update its configuration** (for example, if the port changes or the host moves). The system allows you to modify a service easily.

□ Steps to Modify a Service

1. **Go to the Sidebar**
   - Locate the service under either **Tomcat** or **Windows Services**
2. **Click the Three Vertical Dots (⋮)**
   - Found next to the service name in the sidebar
   - This will open a **context menu**
3. **Select “Edit” or “Modify”**
   - A pop-up titled **“Modify Service”** will appear with pre-filled values
4. **Make the Necessary Changes**
   - Update any of the following fields:

        - **Service Name**: Make sure it still exactly matches the Windows service (if changed)
        - **Service URL**: Host location (e.g., http://localhost, http://192.168.1.100)
        - **Port Number**: The new port the service is running on (e.g., 8005, 8080)

5. **Click the “Modify” Button**
    - This will save your changes and refresh the service with the updated configuration

---

🧪 Example Configuration

Field	Example Value
Service Name	CV PDF
Service URL	http://localhost
Port Number	8005

Once updated, the system will automatically start fetching the new metrics and logs based on the modified values.

6. 🗑️ How to Delete a Service

If a service is no longer needed or was registered incorrectly, you can easily remove it from the **Service Health Monitor** interface.

⚠ **Note:** Only **user-registered Windows services** can be deleted. The **Tomcat Server** is non-removable.

□ Steps to Delete a Service

1. **Locate the Service**
    - Go to the **left sidebar** under **Windows Services**
2. **Click the ⋮ (Three Dots) Icon**

- Located next to the service you want to delete

3. **Select “Delete”**
   - A **confirmation popup** will appear titled **"Confirm Deletion"**
4. **Click “Delete”**
   - Press the red **Delete** button to permanently remove the service from monitoring
5. **To Cancel**
   - Click the **Cancel** button if you change your mind

After deletion, the service will disappear from the sidebar and all its associated logs and metrics will no longer be tracked.

7. ▦ Understanding Top-Right Icons (Header Controls)

At the top-right corner of the dashboard, you’ll find six key icons that provide quick access to core system functions and app controls.

Icon	**Function**
□ **Live**	Indicates that the **Service Health Monitor is running and actively fetching live data**. If this goes off, the system may be paused or disconnected.
● **Power Button**	Used to **Start or Stop individual services** directly from the UI. • Clicking the button will trigger the service to start or stop. • The service status will be reflected in real time via the **Live** indicator and the service tile. ▨ For full instructions, refer to the **“How to Start/Stop a Service”** section. 🐱 **Tomcat Startup Note:** • When you **start Tomcat**, the initial launch process is fast — typically **ready within 10–15 seconds**. • However, **establishing a connection and fetching real-time metrics for the first time** may take **2 to 3 minutes**.

Icon	Function
	• During this period, the service might show as "starting" or partially loaded until all data points (threads, memory, applications, logs) are fully retrieved.
🔔 **Notifications**	Displays all alerts such as: – Service Up/Down – High CPU/Memory Usage – Recovery Events 🔢 The badge count shows unread notifications. 🗑 Use the trash icon to delete all. ✅ Use “Mark all as read” to clear highlights.
🌙 **Theme Mode**	Toggle between **Light Mode** and **Dark Mode** for better readability and user comfort.
⚙ **Settings**	Opens the full **Settings panel** for configuring thresholds, notification preferences, service polling intervals, etc. 🔗 *(Refer to: “Settings” section for detailed guide)*
❓ **Help (Guide)**	Opens this very **User Guide** from anywhere in the dashboard. > *Yes, it's a bit meta – a guide on how to open the guide, but helpful nonetheless!* 😄

8. 🔄 How to Start and Stop a Service

The **Power Button** (located after the **Live** icon and before the **Notification** bell) is used to control the running state of both **Windows** and **Tomcat** services directly from the dashboard.

▶️ Starting a Service

- When a service is **stopped**, the **Power Button** appears in **green** with a **play icon (▶️)**.
- This indicates the service is **ready to be started**.
- Once you click the button:
  - The service startup process will begin.

    - You’ll see a transition to **red**, and the live data panel will update shortly.

⏱ **Tomcat Note:**
Although Tomcat may appear to start within **10–15 seconds**, it may take **2–3 minutes** to fully connect and begin retrieving all metric data (threads, logs, JVM, etc.).

■ Stopping a Service

- When a service is **running**, the **Power Button** turns **red** with a **power icon (⏻)**.
- This means the service is **active and can be stopped**.
- Clicking this button will:
    - Stop the service gracefully
    - Disable metric polling and updates for that service
    - Revert the button back to green (indicating it is now stopped)

This interaction is consistent across all services registered within the application, providing a seamless and intuitive way to manage service availability.

9. ⚙ **Settings**

The **Settings Panel** allows you to configure how the system communicates issues, errors, and status updates. It contains two main sections:

- 🔔 **Notification Preferences**
- 📧 **SMTP Email Configuration**

🔔 Notification Settings

This section lets you control what kinds of notifications are triggered and how they’re delivered.

◈ Notification Methods

Choose how you’d like to receive alerts:

Method	Description
AI Assist	This smart assistant **summarizes technical issues into human-readable messages** and automatically formats them for **in-app**, **email**, and **SMS** notifications. Ideal for non-technical users or managers.
In-app	Displays alerts directly within the dashboard notification bell
Email	Sends notification emails (requires SMTP setup – see below)
SMS	Sends alerts via SMS to configured numbers (requires SMS gateway integration)

✉ Managing Email & SMS Recipients

When **Email** or **SMS** toggles are enabled:

- You will see input fields appear below each section.
- ➕ **Click the plus (+)** button to add:
  - A new email address (e.g., `example@example.com`)
  - A phone number (e.g., `2174141414`)
- 🗑 **Click the trash icon** next to any entry to delete it.
- Multiple contacts can be added and managed independently for email and SMS.

Make sure to hit **Save** after making changes to recipients or toggles.

◈ Service Status Notifications

You can toggle alerts for key service events:

- **Service Down**
- **Service Error**
- **Service Restart**
- **Service Start**

◈ Resource Usage Notifications

Track performance issues in real-time:

- **High CPU Usage**
- **High Memory Usage**

Don’t forget to click the **Save** button after making changes.

📨 SMTP Configuration (For Email Alerts)

To receive email alerts, configure your SMTP settings under the **SMTP Configuration** tab.

Required Fields:

Field	**Description**
SMTP Host	e.g., smtp.office365.com
SMTP Port	e.g., 587 (TLS), 465 (SSL)
Username	Your SMTP email address
Password	Corresponding email password
From Email	Email that will appear as sender
From Name	Display name for the sender
Use SSL	Toggle if your server requires SSL

📩 Click **Test Configuration** to verify your email settings before saving.

10. 🛠 Troubleshooting (For Admins)

If the Service Health Monitor is **not functioning as expected**, follow the steps below to perform a quick diagnosis before reaching out for support.

⚙ Step 1: Check Configuration File

1. **Go to the installation path** of the Service Health Monitor on the server
2. Navigate to the following file:

bash

CopyEdit

backend\.env

3. This .env file contains all critical configuration settings

🔍 Ensure all values in the .env file are correctly defined and not commented out

---

🐱 Tomcat-Specific Troubleshooting

If **Tomcat metrics are not loading** or are behaving abnormally:

✅ 1. Check Tomcat Service Registration

- Open the **Modify Service** popup from the sidebar
- Ensure the following fields are correctly filled:
  - **Service Name**
  - **Service URL (host)**
  - **Port Number**

✅ 2. Validate Tomcat Credentials

- Ensure a **Tomcat username and password** is created in the tomcat-users.xml file
- Add these credentials to your .env file:

env

CopyEdit

```
TOMCAT_USERNAME=your_username

TOMCAT_PASSWORD=your_password
```

✅ 3. Set Tomcat Installation Path

- Also in .env, ensure the full path to the Tomcat installation is correctly defined:

env

CopyEdit

```
TOMCAT_HOME=C:\Path\To\Your\Tomcat
```

📌 Without these credentials and the correct path, the system will fail to fetch logs, threads, and memory details from Tomcat.