ConnectWise PSA Cloud – Service Boards, SLA Behavior, and Correct Configuration

Combined Technical + Operational Master Guide
Version 1.0

Author: Chris
Prepared by: Chris Walker
Date: February 2026

Executive Summary

This document provides a complete, authoritative guide to understanding and configuring **Service Boards**, **SLA behavior**, and **board classification** in the latest version of **ConnectWise PSA Cloud**. It is designed for both:

- **Technical teams** (developers, integrators, architects)
- **Operational teams** (service managers, ConnectWise administrators)

ConnectWise PSA Cloud has changed significantly in recent years, especially around **SLA configuration**, **board behavior**, and **status logic**. This guide clarifies how boards actually work today, how SLA is triggered, and how to configure a board correctly for reactive service operations.

Table of Contents

(Generate automatically in Word after pasting)

---

1. Introduction

ConnectWise PSA Cloud uses a board-based workflow system to manage service tickets, projects, and sales opportunities. However, the platform does **not** expose board type through the API, and the UI has changed significantly, removing the old “Use SLA” checkbox and relocating SLA configuration.

This document explains:

- How to identify the **true board type**
- How SLA works in the **modern cloud UI**
- How to configure a **proper SLA-capable Service Board**
- How to diagnose boards that do not run SLA
- How developers should classify boards using the API

---

2. Understanding Board Types in ConnectWise PSA

2.1 Board Types Overview

ConnectWise has three architectural board types:

- **Service Board** – SLA-capable, reactive work
- **Project Board** – long-running project tasks
- **Opportunity Board** – sales pipeline stages

These types are **not exposed as a field** in the UI or API.

2.2 How Board Type Is Determined

A board's type is determined **solely by which setup table it appears under**:

Setup Table	Board Type
System → Setup Tables → **Service Board**	Service Board
System → Setup Tables → **Project Board**	Project Board
System → Setup Tables → **Opportunity Board**	Sales Board

If a board appears under Service Board, it is a **true Service Board**.

---

3. SLA Behavior in the Latest PSA Cloud

3.1 SLA Configuration Has Moved

In the latest cloud UI, the old **SLA tab on boards has been removed**.

SLA is now configured under:

System → Setup Tables → Service Desk → SLA

This includes:

- SLA rules
- SLA timers
- SLA priorities
- SLA exclusions

3.2 SLA Is Now Implicit

There is no longer a “Use SLA” checkbox.

SLA is active when:

1. The board is a **Service Board**
2. The ticket’s **status** has the correct SLA flags
3. The ticket’s **priority** has SLA rules
4. The ticket’s **Type/Subtype/Item** is not excluded

If any of these conditions fail, SLA will not run.

---

4. SLA-Related Status Flags

Statuses on a Service Board may include the following SLA flags:

- **Needs Response**
- **Stops Clock**
- **Closes Ticket**

These flags determine SLA timing behavior.

4.1 Where to Find Status Flags

Navigate to:

System → Setup Tables → Service Board → [Board] → Statuses

Each status may include SLA behavior flags.

4.2 What Each Flag Means

Flag	Meaning
Needs Response	SLA RespondedDate starts here
Stops Clock	SLA pauses
Closes Ticket	SLA ends

4.3 Workflow-Only Boards

Boards with statuses like:

- “We have responded”
- “We have created a plan”
- “We are waiting”
- “We have resolved the issue”

…are **workflow boards**, not SLA boards.
These statuses do **not** include SLA flags and will never trigger SLA timers.

---

5. Correct Configuration of a Service Board

This section provides the recommended configuration for a fully SLA-capable Service Board.

5.1 Step 1 — Create or Verify the Board

Go to:

System → Setup Tables → Service Board

Ensure the board exists here.

5.2 Step 2 — Configure SLA-Aware Statuses

Recommended statuses:

Status	SLA Flag
New	Needs Response
In Progress	Stops Clock (optional)
Waiting on Customer	Stops Clock
Completed	Closes Ticket
Closed	Closes Ticket

5.3 Step 3 — Configure SLA Rules

Go to:

System → Setup Tables → Service Desk → SLA

Configure:

- Responded time

- Resolved time
- Priority-based timers
- Business hours
- Escalation rules

5.4 Step 4 — Check SLA Exclusions

Ensure the ticket's:

- Type
- Subtype
- Item

…are not excluded.

5.5 Step 5 — Ensure Priorities Match SLA Rules

If a priority has no SLA rule, SLA will not run.

---

6. Diagnosing SLA Issues

Use this checklist to diagnose SLA problems.

6.1 SLA Diagnostic Checklist

Check	Result
Board appears under Service Board	Required
Status has Needs Response	Required for RespondedDate
Priority has SLA rules	Required
T/S/I not excluded	Required
Ticket enters a Needs Response status	Required

If any item is missing → SLA will not run.

---

7. Developer Guide: Determining Board Type and SLA Capability via API

7.1 Determining Board Type via API

Because the API does not expose board type, developers must infer it:

- If returned by `/service/boards` → **Service Board**
- If returned by `/project/boards` → **Project Board**
- If returned by `/sales/boards` → **Sales Board**

7.2 Checking SLA Capability

A board is SLA-capable if:

- It is a Service Board
- `useSla` is true (from `/service/boards/{id}`)
- Statuses include SLA flags

7.3 Example C# Logic

```
if (boardType == Service && serviceBoard.UseSla)
{
    // SLA-capable
}
```

---

8. Best Practices for Service Board Design

- Keep SLA boards simple
- Avoid conversational workflow statuses
- Use consistent SLA flags
- Ensure priorities match SLA rules
- Avoid mixing project-like work on service boards
- Review SLA exclusions regularly

---

9. Appendix A — Example DTO for `/service/boards/{id}`

```
public class ServiceBoardDto
{
    public int Id { get; set; }
    public string Name { get; set; }
    public bool UseSla { get; set; }
    public bool ProjectsFlag { get; set; }
    public int? TypeId { get; set; }
    public string Category { get; set; }
    public int? LocationId { get; set; }
    public int? DepartmentId { get; set; }
    public int? BusinessUnitId { get; set; }
    public int? SortOrder { get; set; }
    public bool? ClosedFlag { get; set; }
    public bool? DefaultFlag { get; set; }

    [JsonIgnore]
    public object _info { get; set; }
```

```
}
```

---

10. Appendix B — Board Classification Logic (C#)

```
public enum BoardType
{
    Service,
    Project,
    Sales,
    Unknown
}
```
