Rules Engine User Guide

This guide walks through the complete workflow for creating, deploying, and verifying compliance rule packs in MyTeacher.

Table of Contents

- Overview
- Data Flow Diagram
- Step 1: Planning Your Rule Pack
- Step 2: Creating a Rule Pack
- Step 3: Adding Rules to the Pack
- Step 4: Configuring Evidence Requirements
- Step 5: Activating the Rule Pack
- Step 6: Verifying Deployment
- Step 7: Monitoring Usage
- Troubleshooting
- Complete Example: Maryland IEP Rules

Overview

The rules engine enforces compliance requirements for IEP, 504, and BIP meetings. Rule packs define:

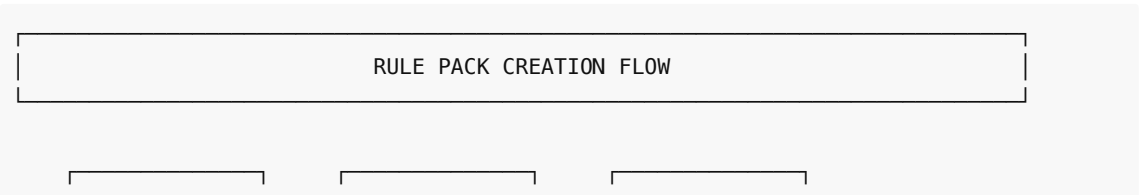
- **Document delivery deadlines** (pre/post meeting)
- **Required evidence** (conference notes, consent forms)
- **Enforcement gates** (consent before implementation)
- **Recording policies** (staff must record if parent records)

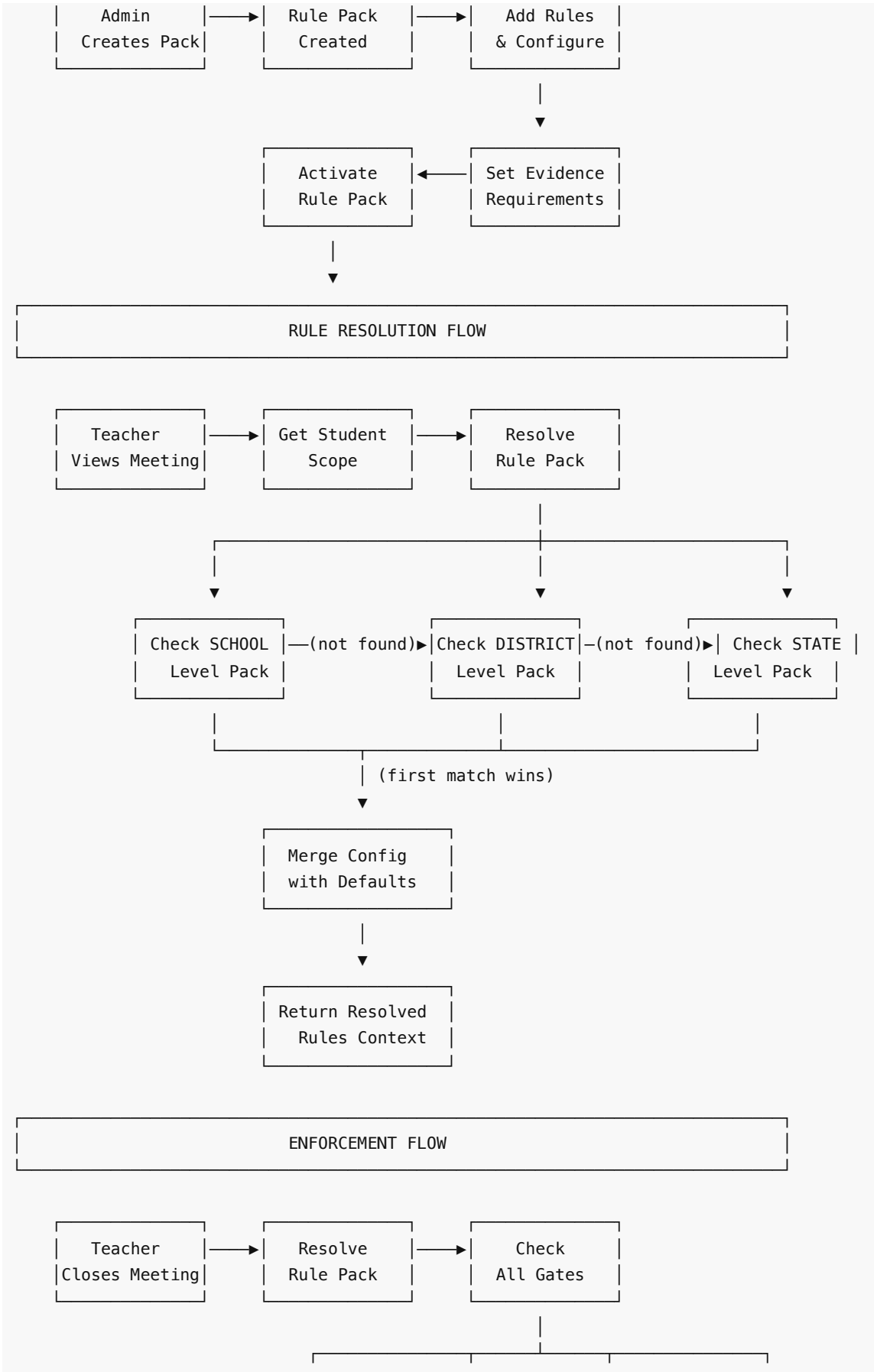
Key Concepts

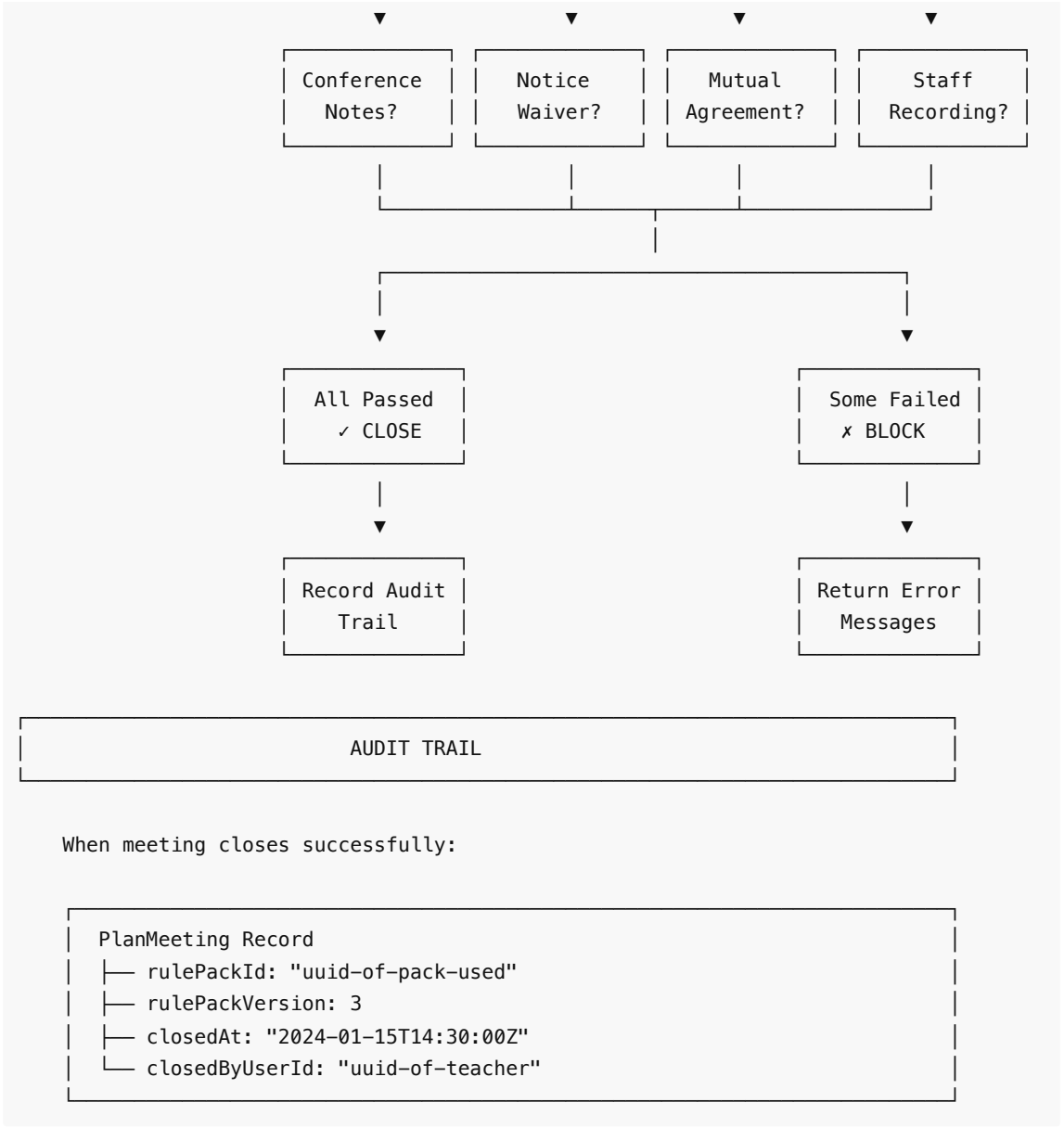
Concept	Description
Rule Pack	A collection of rules scoped to a jurisdiction (state/district/school) and plan type
Scope Precedence	SCHOOL > DISTRICT > STATE (more specific overrides general)
Rule Definition	A template rule that can be added to packs with custom configuration
Evidence Type	Documents or acknowledgments that can be required for meetings
Enforcement	Rules are checked when closing meetings or implementing plans

---

Data Flow Diagram




Admin Creates Pack
Rule Pack Created
Add Rules & Configure
Set Evidence Requirements
Activate Rule Pack
RULE RESOLUTION FLOW
Teacher Views Meeting
Get Student Scope
Resolve Rule Pack
Check SCHOOL Level Pack
(not found)
Check DISTRICT Level Pack
(not found)
Check STATE Level Pack
(first match wins)
Merge Config with Defaults
Return Resolved Rules Context
ENFORCEMENT FLOW
Teacher Closes Meeting
Resolve Rule Pack
Check All Gates



Step 1: Planning Your Rule Pack

Before creating a rule pack, determine:

1.1 Scope Level

Scope	Use When	Example
STATE	Default rules for entire state	Maryland IEP rules
DISTRICT	District-specific overrides	Howard County overrides
SCHOOL	School-specific overrides	Wilde Lake HS exceptions

1.2 Plan Type

Plan Type	Description
IEP	Individualized Education Program
PLAN504	Section 504 Accommodation Plan
BIP	Behavior Intervention Plan
ALL	Applies to all plan types

1.3 Rule Configuration

Review available rules and their defaults:

Rule	Default	Description
PRE_MEETING_DOCS_DAYS	5 days	Business days before meeting for draft delivery
POST_MEETING_DOCS_DAYS	5 days	Business days after meeting for final delivery
US_MAIL_PRE_MEETING_DAYS	3 days	Extra days for US Mail pre-meeting
US_MAIL_POST_MEETING_DAYS	3 days	Extra days for US Mail post-meeting
CONFERENCE_NOTES_REQUIRED	true	Require notes before closing
INITIAL_IEP_CONSENT_GATE	true	Block implementation without consent
CONTINUED_MEETING_NOTICE_DAYS	10 days	Minimum notice for continued meetings
CONTINUED_MEETING_MUTUAL_AGREEMENT	true	Require agreement documentation
AUDIO_RECORDING_RULE	staff records if parent does	Recording policy

---

Step 2: Creating a Rule Pack

Via Admin UI (Recommended)

1. Navigate to `/admin/rules`
2. Click **"New Rule Pack"** button
3. Fill in the form:

Field	Example Value	Notes
Scope Type	STATE	Choose STATE, DISTRICT, or SCHOOL

Scope ID	MD	State code, district ID, or school ID
Plan Type	IEP	Which plan type this applies to
Name	Maryland State IEP Rules	Descriptive name
Effective From	2024-01-01	When rules take effect
Effective To	(leave blank)	Optional end date

4. Click **Create**

Via API

```
curl -X POST https://your-app.vercel.app/api/admin/rule-packs \
  -H "Content-Type: application/json" \
  -H "Cookie: connect.sid=your-session-cookie" \
  -d '{
    "scopeType": "STATE",
    "scopeId": "MD",
    "planType": "IEP",
    "name": "Maryland State IEP Rules",
    "effectiveFrom": "2024-01-01",
    "isActive": false
  }'
```

Response:

```
{
  "id": "abc123-uuid",
  "scopeType": "STATE",
  "scopeId": "MD",
  "planType": "IEP",
  "name": "Maryland State IEP Rules",
  "version": 1,
  "isActive": false,
  "effectiveFrom": "2024-01-01T00:00:00.000Z",
  "effectiveTo": null,
  "createdAt": "2024-01-15T10:00:00.000Z"
}
```

Step 3: Adding Rules to the Pack

Via Admin UI

1. Select the rule pack from the list
2. Click the **"Rules"** tab
3. For each rule you want to enable:
   - Toggle the rule **ON**
   - Configure parameters (e.g., days = 7)

4. Click **Save Changes**

Via API

First, get the list of available rule definitions:

```
curl https://your-app.vercel.app/api/rule-packs/definitions \
  -H "Cookie: connect.sid=your-session-cookie"
```

Then add rules to your pack:

```
# Add PRE_MEETING_DOCS_DAYS rule with 7 days
curl -X POST https://your-app.vercel.app/api/admin/rule-packs/{packId}/rules \
  -H "Content-Type: application/json" \
  -H "Cookie: connect.sid=your-session-cookie" \
  -d '{
    "ruleDefinitionId": "uuid-of-pre-meeting-docs-rule",
    "isEnabled": true,
    "config": { "days": 7 },
    "sortOrder": 1
  }'
```

Bulk Update (Recommended)

```
curl -X PUT https://your-app.vercel.app/api/admin/rule-packs/{packId}/rules/bulk \
  -H "Content-Type: application/json" \
  -H "Cookie: connect.sid=your-session-cookie" \
  -d '{
    "rules": [
      {
        "ruleDefinitionId": "uuid-pre-meeting",
        "isEnabled": true,
        "config": { "days": 7 }
      },
      {
        "ruleDefinitionId": "uuid-post-meeting",
        "isEnabled": true,
        "config": { "days": 5 }
      },
      {
        "ruleDefinitionId": "uuid-conference-notes",
        "isEnabled": true,
        "config": { "required": true }
      }
    ]
  }'
```

Step 4: Configuring Evidence Requirements

Evidence requirements specify what documentation is needed for each rule.

Via Admin UI

1. Select the rule pack
2. Click the **"Evidence"** tab
3. For each rule, select required evidence types:
   - Conference Notes
   - Parent Consent Form
   - Notice Waiver
   - Recording Acknowledgment
   - Pre/Post Meeting Docs Confirmation
4. Toggle **Required** for mandatory evidence
5. Click **Save Changes**

Via API

```
# Add CONFERENCE_NOTES as required evidence for a rule
curl -X POST https://your-app.vercel.app/api/admin/rule-
packs/{packId}/rules/{ruleId}/evidence \
  -H "Content-Type: application/json" \
  -H "Cookie: connect.sid=your-session-cookie" \
  -d '{
    "evidenceTypeId": "uuid-of-conference-notes-type",
    "isRequired": true
  }'
```

Step 5: Activating the Rule Pack

Important: Only ONE rule pack can be active per (scopeType, scopeId, planType) combination.

Via Admin UI

1. Select the rule pack
2. On the **Overview** tab, toggle **Active** to ON
3. Confirm activation

Via API

```
curl -X PATCH https://your-app.vercel.app/api/admin/rule-packs/{packId} \
  -H "Content-Type: application/json" \
  -H "Cookie: connect.sid=your-session-cookie" \
  -d '{
    "isActive": true
  }'
```

Deactivating Another Pack First

If another pack is already active:

```
# Deactivate existing pack
curl -X PATCH https://your-app.vercel.app/api/admin/rule-packs/{oldPackId} \
  -H "Content-Type: application/json" \
  -H "Cookie: connect.sid=your-session-cookie" \
  -d '{
    "isActive": false
  }'

# Then activate new pack
curl -X PATCH https://your-app.vercel.app/api/admin/rule-packs/{newPackId} \
  -H "Content-Type: application/json" \
  -H "Cookie: connect.sid=your-session-cookie" \
  -d '{
    "isActive": true
  }'
```

Step 6: Verifying Deployment

6.1 Check Rule Pack Status

Admin UI:

- Go to `/admin/rules`
- Verify pack shows green **Active** badge
- Check the **Preview** tab shows merged configuration

API:

```
curl https://your-app.vercel.app/api/admin/rule-packs/{packId} \
  -H "Cookie: connect.sid=your-session-cookie"
```

Expected response includes:

```
{
  "id": "abc123",
  "isActive": true,
  "rules": [...],
  "version": 1
}
```

6.2 Test Rule Resolution

Test that rules resolve correctly for a student:

```
curl "https://your-app.vercel.app/api/rules/context?studentId=
{studentId}&planType=IEP" \
  -H "Cookie: connect.sid=your-session-cookie"
```

Evidence of successful deployment:

```
{
  "resolved": true,
  "rulePack": {
    "id": "abc123",
    "name": "Maryland State IEP Rules",
    "version": 1,
    "scopeType": "STATE",
    "scopeId": "MD",
    "planType": "IEP"
  },
  "precedence": {
    "searched": [
      { "scopeType": "SCHOOL", "scopeId": "school-id" },
      { "scopeType": "DISTRICT", "scopeId": "district-id" },
      { "scopeType": "STATE", "scopeId": "MD" }
    ],
    "matched": { "scopeType": "STATE", "scopeId": "MD" }
  },
  "gates": [
    {
      "key": "CONFERENCE_NOTES_REQUIRED",
      "name": "Conference Notes Required",
      "enabled": true,
      "config": { "required": true }
    }
  ],
  "deadlines": {
    "preMeetingDocs": {
      "standardDeadline": "2024-01-08",
      "usMailDeadline": "2024-01-03",
      "businessDays": 7
    }
  },
  "evidenceRequirements": [
    {
      "key": "CONFERENCE_NOTES",
      "name": "Conference Notes",
      "isRequired": true
    }
  ]
}
```


Key indicators:

- `resolved: true` - A pack was found
- `rulePack.name` - Shows which pack matched
- `precedence.matched` - Shows the scope level that matched
- `gates` - Shows active enforcement rules
- `deadlines` - Shows calculated due dates

6.3 Verify No Pack Scenario

If no pack is configured for a scope:

```
{
  "resolved": false,
  "rulePack": null,
  "precedence": {
    "searched": [...],
    "matched": null
  },
  "gates": [],
  "evidenceRequirements": []
}
```

Step 7: Monitoring Usage

7.1 Meeting Audit Trail

Every closed meeting records which rule pack was used:

```
SELECT
  pm.id,
  pm."studentId",
  pm."planType",
  pm."rulePackId",
  pm."rulePackVersion",
  pm."closedAt",
  pm."closedByUserId",
  rp.name as "rulePackName"
FROM "PlanMeeting" pm
LEFT JOIN "RulePack" rp ON pm."rulePackId" = rp.id
WHERE pm."closedAt" IS NOT NULL
ORDER BY pm."closedAt" DESC
LIMIT 50;
```

7.2 Enforcement Failures

Track when meetings were blocked from closing:

Application logs will show:

```
[ENFORCEMENT] Meeting {meetingId} close blocked: MISSING_CONFERENCE_NOTES
[ENFORCEMENT] Meeting {meetingId} close blocked: MISSING_CONSENT
```

7.3 Rule Pack Usage Statistics

```
-- Count meetings closed per rule pack
SELECT
  rp.name,
  rp."scopeType",
  rp."scopeId",
```

```
  rp.version,
  COUNT(pm.id) as "meetingsClosed"
FROM "RulePack" rp
LEFT JOIN "PlanMeeting" pm ON pm."rulePackId" = rp.id
WHERE pm."closedAt" IS NOT NULL
GROUP BY rp.id, rp.name, rp."scopeType", rp."scopeId", rp.version
ORDER BY "meetingsClosed" DESC;
```

7.4 Evidence Collection Statistics

```
-- Evidence collected per meeting
SELECT
  pm.id as "meetingId",
  pm."closedAt",
  COUNT(me.id) as "evidenceCount",
  STRING_AGG(ret.key, ', ') as "evidenceTypes"
FROM "PlanMeeting" pm
JOIN "MeetingEvidence" me ON me."meetingId" = pm.id
JOIN "RuleEvidenceType" ret ON ret.id = me."evidenceTypeId"
WHERE pm."closedAt" IS NOT NULL
GROUP BY pm.id, pm."closedAt"
ORDER BY pm."closedAt" DESC;
```

7.5 Prisma Studio

For visual database exploration:

```
pnpm --filter @myteacher/db exec prisma studio
```

Browse these tables:

- **RulePack** - View all packs and their status
- **RulePackRule** - See rules linked to packs
- **RulePackEvidenceRequirement** - View evidence requirements
- **PlanMeeting** - Check `rulePackId` and `rulePackVersion` columns

Troubleshooting

Rule Pack Not Found

Symptom: `resolved: false` in rules context response

Checklist:

1. Is there a pack for this scope (state/district/school)?
2. Is `isActive: true`?
3. Is `effectiveFrom <= today`?
4. Is `effectiveTo` null or in the future?
5. Does the pack match the plan type (IEP/504/BIP)?

Wrong Rule Pack Applied

Symptom: Different pack than expected is matched

Check precedence:

```
curl "https://your-app.vercel.app/api/rules/context?studentId=
{studentId}&planType=IEP"
```

Review the `precedence.searched` array - the first active pack found wins.

Evidence Not Required

Symptom: Meeting closes without expected evidence

Check:

1. Is the evidence type linked to the rule?
2. Is `isRequired: true` for that evidence?
3. Is the rule itself enabled (`isEnabled: true`)?

Cannot Activate Pack

Symptom: Error when activating

Cause: Another pack is already active for the same scope+planType

Solution: Deactivate the existing pack first, then activate the new one.

Complete Example: Maryland IEP Rules

Step-by-Step Setup

1. Create the Pack

```
curl -X POST https://your-app.vercel.app/api/admin/rule-packs \
  -H "Content-Type: application/json" \
  -H "Cookie: connect.sid=your-session" \
  -d '{
    "scopeType": "STATE",
    "scopeId": "MD",
    "planType": "IEP",
    "name": "Maryland State IEP Compliance Rules",
    "effectiveFrom": "2024-01-01",
    "isActive": false
  }'
```

Save the returned `id` (e.g., `pack-uuid-123`).

2. Get Rule Definitions

```
curl https://your-app.vercel.app/api/rule-packs/definitions \
```

```
  -H "Cookie: connect.sid=your-session"
```

Note the UUIDs for each rule definition.

3. Add Rules with Maryland-Specific Config

```
curl -X PUT https://your-app.vercel.app/api/admin/rule-packs/pack-uuid-
123/rules/bulk \
  -H "Content-Type: application/json" \
  -H "Cookie: connect.sid=your-session" \
  -d '{
    "rules": [
      {
        "ruleDefinitionId": "uuid-pre-meeting-docs",
        "isEnabled": true,
        "config": { "days": 5 }
      },
      {
        "ruleDefinitionId": "uuid-post-meeting-docs",
        "isEnabled": true,
        "config": { "days": 5 }
      },
      {
        "ruleDefinitionId": "uuid-conference-notes",
        "isEnabled": true,
        "config": { "required": true }
      },
      {
        "ruleDefinitionId": "uuid-consent-gate",
        "isEnabled": true,
        "config": { "enabled": true }
      },
      {
        "ruleDefinitionId": "uuid-continued-notice",
        "isEnabled": true,
        "config": { "days": 10 }
      },
      {
        "ruleDefinitionId": "uuid-mutual-agreement",
        "isEnabled": true,
        "config": { "required": true }
      },
      {
        "ruleDefinitionId": "uuid-audio-recording",
        "isEnabled": true,
        "config": {
          "staffMustRecordIfParentRecords": true,
          "markAsNotOfficialRecord": true
        }
      }
```

```
    ]
  }'
```

4. Add Evidence Requirements

For the conference notes rule:

```
curl -X POST https://your-app.vercel.app/api/admin/rule-packs/pack-uuid-
123/rules/{conferenceNotesRuleId}/evidence \
  -H "Content-Type: application/json" \
  -H "Cookie: connect.sid=your-session" \
  -d '{
    "evidenceTypeId": "uuid-conference-notes-type",
    "isRequired": true
  }'
```

5. Activate the Pack

```
curl -X PATCH https://your-app.vercel.app/api/admin/rule-packs/pack-uuid-123 \
  -H "Content-Type: application/json" \
  -H "Cookie: connect.sid=your-session" \
  -d '{
    "isActive": true
  }'
```

6. Verify Deployment

```
# Test with any Maryland student
curl "https://your-app.vercel.app/api/rules/context?studentId=
{mdStudentId}&planType=IEP" \
  -H "Cookie: connect.sid=your-session"
```

Expected:

```
{
  "resolved": true,
  "rulePack": {
    "name": "Maryland State IEP Compliance Rules",
    "scopeType": "STATE",
    "scopeId": "MD"
  },
  "precedence": {
    "matched": { "scopeType": "STATE", "scopeId": "MD" }
  }
}
```

7. Monitor Usage

After teachers start using the system, query the audit trail:

```sql
SELECT
  pm."closedAt",
  s."firstName" || ' ' || s."lastName" as student,
  u."firstName" || ' ' || u."lastName" as "closedBy",
  rp.name as "rulePackUsed",
  rp.version
FROM "PlanMeeting" pm
JOIN "Student" s ON s.id = pm."studentId"
JOIN "User" u ON u.id = pm."closedByUserId"
JOIN "RulePack" rp ON rp.id = pm."rulePackId"
WHERE pm."rulePackId" = 'pack-uuid-123'
ORDER BY pm."closedAt" DESC;
```

Summary Checklist

Use this checklist when deploying new rule packs:

- [ ] **Plan**: Determine scope, plan type, and rule configurations
- [ ] **Create**: Create rule pack via UI or API
- [ ] **Configure**: Add rules with appropriate settings
- [ ] **Evidence**: Set required evidence for each rule
- [ ] **Activate**: Toggle pack to active (deactivate old pack first if needed)
- [ ] **Verify**: Test with `/api/rules/context` - confirm `resolved: true`
- [ ] **Test**: Close a test meeting to verify enforcement works
- [ ] **Monitor**: Check audit trail for `rulePackId` on closed meetings
- [ ] **Document**: Record which pack is active for each scope