Processing Policies Specification

Version: 1.0
Status: 🚧 Draft
Date: 2026-02-26
Author: CaC-ConfigMgr Product Team

1. Executive Summary

Processing Policies (PP) are **configuration resources** that link together:

- **1 Routing Policy** (where to store)
- **1 Normalization Policy** (how to parse)
- **1 Enrichment Policy** (what context to add)

```
Processing Policy = RP + NP + EP

Log Source → [Processing Policy] → Storage
                    │
                    ├── RP: routing-policy-ref → selects repo
                    ├── NP: normalization-policy-ref → parses log
                    └── EP: enrichment-policy-ref → adds context
```

Role: Simplify configuration by grouping 3 policies into 1 reference.

2. Structure

2.1 YAML Definition

```
# templates/mssp/acme-corp/base/processing-policies.yaml
apiVersion: cac-configmgr.io/v1
kind: ConfigTemplate
metadata:
  name: acme-base
  extends: logpoint/golden-base

spec:
  processingPolicies:
    - name: windows-security-pipeline
      _id: pp-windows-sec

      # References to the 3 policies
      routingPolicy: rp-windows-security
      normalizationPolicy: np-windows
      enrichmentPolicy: ep-geoip-threatintel
      # Metadata
      description: "Complete pipeline for Windows security logs"
      enabled: true
```

2.2 Fields

Field	Type	Description	Required
`name`	string	Unique PP name	✅ Yes
`_id`	string	Template ID for inheritance	✅ Yes
`routingPolicy`	string	RoutingPolicy reference	✅ Yes
`normalizationPolicy`	string	NormalizationPolicy reference	× No (default: Auto)
`enrichmentPolicy`	string	EnrichmentPolicy reference	× No
`description`	string	Description	× No
`enabled`	bool	Active/inactive	× No (default: true)

---

3. Inheritance

Same mechanism as other resources: `_id` for matching.

```
# Parent: logpoint/golden-base/processing-policies.yaml
spec:
  processingPolicies:
    - name: default-pipeline
      _id: pp-default
      routingPolicy: rp-default
      normalizationPolicy: np-auto
      enrichmentPolicy: ep-basic

# Child: mssp/acme-corp/base/processing-policies.yaml
spec:
  processingPolicies:
    - name: default-pipeline
      _id: pp-default
      routingPolicy: rp-acme-default        # Override
      # normalizationPolicy: inherited (np-auto)
      enrichmentPolicy: ep-acme-geoip       # Override
```

---

4. Examples

4.1 LogPoint Golden Template

```yaml
# templates/logpoint/golden-base/processing-policies.yaml
spec:
  processingPolicies:
    - name: default
      _id: pp-default
      routingPolicy: rp-default
      normalizationPolicy: np-auto

    - name: windows-security
      _id: pp-windows-sec
      routingPolicy: rp-windows
      normalizationPolicy: np-windows
      enrichmentPolicy: ep-geoip

    - name: linux-syslog
      _id: pp-linux
      routingPolicy: rp-linux
      normalizationPolicy: np-syslog

    - name: firewall-perimeter
      _id: pp-firewall
      routingPolicy: rp-security
      normalizationPolicy: np-common-firewall
      enrichmentPolicy: ep-threat-intel
```

4.2 MSSP Extension

```yaml
# templates/mssp/acme-corp/base/processing-policies.yaml
spec:
  processingPolicies:
    - name: default
      _id: pp-default
      routingPolicy: rp-acme-default
      normalizationPolicy: np-auto
      enrichmentPolicy: ep-acme-geoip       # Add custom GeoIP

    - name: windows-security
      _id: pp-windows-sec
      routingPolicy: rp-acme-windows        # Override routing
      normalizationPolicy: np-windows
      enrichmentPolicy: ep-acme-full        # GeoIP + ThreatIntel + AD

    - name: high-value-assets
      _id: pp-high-value
      routingPolicy: rp-critical-assets
      normalizationPolicy: np-auto
      enrichmentPolicy: ep-premium          # All enrichments
```

4.3 Instance

```yaml
# instances/client-bank/prod/instance.yaml
spec:
  processingPolicies:
    - name: windows-security
      _id: pp-windows-sec
      routingPolicy: rp-bank-windows        # Override: bank-specific routing
      # normalizationPolicy: inherited
      # enrichmentPolicy: inherited
```

5. Usage

5.1 Association with Devices

Devices reference the PP to use:

```yaml
# devices.yaml
devices:
  - name: windows-dc-01
    type: windows-wec
    processingPolicy: pp-windows-sec       # ← References the PP

  - name: firewall-checkpoint-01
    type: checkpoint
    processingPolicy: pp-firewall

  - name: linux-server-generic
    type: syslog
    processingPolicy: pp-default
```

5.2 Benefits

- **Simplicity**: 1 reference instead of 3
- **Consistency**: Ensures RP/NP/EP are compatible
- **Inheritance**: Change entire pipeline in 1 place

6. Validation

6.1 Rules

Rule	Severity	Description
`routingPolicy` exists	ERROR	Must reference an existing RP
`normalizationPolicy` exists	ERROR	If specified, must exist
`enrichmentPolicy` exists	ERROR	If specified, must exist

Rule	Severity	Description
No cycle	ERROR	EP must not reference PP (indirect)

6.2 Example Error

```
# INVALID: Non-existent reference
processingPolicies:
  - name: bad-pipeline
    routingPolicy: rp-inexistent           # ERROR: RP not defined
    normalizationPolicy: np-windows
```

Appendix: Quick Reference

```
processingPolicies:
  - name: <string>              # Required
    _id: <string>               # For inheritance
    routingPolicy: <string>     # Required → RoutingPolicy
    normalizationPolicy: <string> # Optional → NormalizationPolicy
    enrichmentPolicy: <string>    # Optional → EnrichmentPolicy
    description: <string>
    enabled: <bool>
```

Open Questions

1. **Optionals**: Are `normalizationPolicy` and `enrichmentPolicy` really optional?
2. **Defaults**: Default values if not specified (`np-auto`, no EP)?
3. **Reusability**: Can a PP be used by multiple devices? (Yes, that's the point)