

Hierarchical Template System Specification

Version: 1.0

Status: Draft

Date: 2026-02-26

Author: CaC-ConfigMgr Product Team

1. Executive Summary

CaC-ConfigMgr CaC implements a **multi-level hierarchical template system** that enables:

- **LogPoint** to provide battle-tested baseline configurations (Golden Templates)
- **MSSPs** to customize and extend baselines for their operational needs
- **End Customers** to instantiate specific configurations with environment-specific values

This system eliminates configuration duplication while ensuring consistency, compliance, and scalability across thousands of SIEM deployments.

2. Problem Statement

Current State (DirSync & Manual)

- Each client configuration is a standalone copy
- No inheritance: changing a standard requires updating N files
- Risk of configuration drift between similar clients
- No standardized compliance baselines from vendor

Target State (CaC-ConfigMgr CaC)

- **Single source of truth:** Base templates maintained by LogPoint
 - **Progressive specialization:** MSSP → Client Type → Instance
 - **Automatic propagation:** Update base = update all children
 - **Compliance by default:** Golden templates enforce security best practices
-

3. Core Concepts

3.1 Template Hierarchy Levels

```
LEVEL 1: LOGPOINT GOLDEN (Provider)
├── Vendor-maintained
├── Security-hardened defaults
└── API-optimized configurations
    ↓ extends
```

```
LEVEL 2: MSSP BASE (Organization)
```

```

    └── MSSP-specific standards
    └── Infrastructure conventions (mount points, naming)
        └── Compliance overlays
            ↓ extends

LEVEL 3: DEPLOYMENT PROFILE (Pattern)
└── Simple: Single AIO, basic retention
└── Medium: Distributed, standard rotation
└── Enterprise: Multi-cluster, complex retention policies
└── Custom: MSSP-defined patterns
    ↓ extends

LEVEL 4: INSTANCE (Concrete)
└── Environment-specific values (prod/staging/dev)
└── Customer-specific exceptions
└── Deployment targets (Fleet references)

```

3.2 Inheritance Mechanisms

Mechanism	Description	Use Case
Inherit	Copy all resources from parent unchanged	Standard compliance baselines
Override	Replace resource with same identifier	Change retention, modify paths
Append	Add new resources not in parent	Additional repos, custom policies
Patch	Partial modification of parent resource	Add a routing criteria, extend list
Delete	Remove resource from parent (explicit)	Disable default repo for specific case

3.3 Resource Identification (Top Level)

Top-level resources are identified by their **kind + name** tuple:

```

kind: RepoConfig
name: repo-secu # ← Unique identifier for inheritance

```

Same **name** in child = **Merge/Patch** (see 3.4)

Different **name** in child = **Append**

3.4 Template IDs for List Matching

When resources contain **lists** (e.g., `hiddenrepopath`, `routing_criteria`), elements are matched using **_id** fields:

```

repos:
  - name: repo-secu
    hiddenrepopath:
      - _id: fast-tier      # ← Template ID (internal, not sent to API)

```

```

    path: /opt/immune/storage
    retention: 30

    - _id: warm-tier      # ← Template ID for second element
      path: /opt/immune/storage-warm
      retention: 90
  
```

Matching Rules:

Scenario	Result
Same <code>_id</code> in parent and child	Merge: Child fields override parent, undefined fields inherited
<code>_id</code> only in parent	Inherit: Element kept as-is
<code>_id</code> only in child	Append: New element added to list
<code>_action: delete</code> on existing <code>_id</code>	Remove: Element removed from list

Important: `_id` fields are **internal to CaC-ConfigMgr** and are **filtered out** before sending to LogPoint API.

4. Syntax Specification

4.1 Template Definition

```

# templates/logpoint/golden-base.yaml
apiVersion: cac-configmgr.io/v1
kind: ConfigTemplate
metadata:
  name: golden-base
  version: "2.1.0"
  provider: logpoint

spec:
  # Resources are grouped by type
  repos:
    - name: default
      hiddenrepopath:
        - _id: default-tier
          path: /opt/immune/storage
          retention: 90

  routingPolicies:
    - policy_name: rp-default
      catch_all: default
      routing_criteria: []

  normalizationPolicies:
    - policy_name: np-auto
      normalization_packages: ["Auto"]
  
```

4.2 Template Extension with Merge

```
# templates/mssp/acme-corp/base.yaml
apiVersion: cac-configmgr.io/v1
kind: ConfigTemplate
metadata:
  name: acme-base
  extends: logpoint/golden-base # ← Parent reference
  version: "1.0.0"
  provider: acme-mssp

spec:
  # MERGE: Same name, same _id = partial override
  repos:
    - name: default
      hiddenrepopath:
        - _id: default-tier          # ← Same _id as parent
          retention: 180            # ← Only override retention
          # path inherited: /opt/immune/storage

  # APPEND: New name = add to catalog
  - name: repo-secu
    hiddenrepopath:
      - _id: fast-tier
        path: /opt/immune/storage
        retention: 365

    - name: repo-archive
      hiddenrepopath:
        - _id: warm-tier
          path: /opt/immune/storage-warm
          retention: 90
        - _id: cold-tier
          path: /opt/immune/storage-cold
          retention: 1095
```

Result after merge:

```
repos:
  - name: default
    hiddenrepopath:
      - _id: default-tier
        path: /opt/immune/storage          # ← Inherited from parent
        retention: 180                   # ← Override by child

  - name: repo-secu # ← Added
    hiddenrepopath:
      - _id: fast-tier
        path: /opt/immune/storage
        retention: 365
```

```

- name: repo-archive # ← Added
  hiddenrepopath:
    - _id: warm-tier
      path: /opt/immune/storage-warm
      retention: 90
    - _id: cold-tier
      path: /opt/immune/storage-cold
      retention: 1095
  
```

4.3 Multi-Level Inheritance with List Merging

```

# templates/mssp/acme-corp/profiles/enterprise.yaml
apiVersion: cac-configmgr.io/v1
kind: ConfigTemplate
metadata:
  name: acme-enterprise
  extends: acme-corp/base # ← Can extend non-golden templates

spec:
  repos:
    # Merge: Override retention, add new tiers
    - name: repo-secu
      hiddenrepopath:
        - _id: fast-tier # ← Matches parent's fast-tier
          retention: 7 # ← Override: was 365
          # path inherited: /opt/immune/storage

        - _id: warm-tier # ← New _id = append
          path: /opt/immune/storage-warm
          retention: 90

        - _id: nfs-tier # ← New _id = append
          path: /opt/immune/storage-nfs
          retention: 3650

    # Append enterprise-specific
    - name: repo-trading
      hiddenrepopath:
        - _id: fast-tier
          path: /opt/immune/storage
          retention: 2
        - _id: warm-tier
          path: /opt/immune/storage-warm
          retention: 30
  
```

Chain resolution:

```

LogPoint (golden-base)
└─ repo-secu: [fast-tier: /opt/immune/storage, 365d]

    ↓ extends

ACME Base (acme-base)
└─ repo-secu: [fast-tier: 365d] ← Merge: only changed retention

    ↓ extends

Enterprise (acme-enterprise)
└─ repo-secu:
    - fast-tier: retention=7      ← Override again
    - warm-tier: added           ← New element
    - nfs-tier: added            ← New element

Final resolved:
repo-secu:
- _id: fast-tier
  path: /opt/immune/storage      ← From LogPoint (inherited)
  retention: 7                  ← From Enterprise (override)
- _id: warm-tier
  path: /opt/immune/storage-warm
  retention: 90
- _id: nfs-tier
  path: /opt/immune/storage-nfs
  retention: 3650

```

4.4 Instance Instantiation

```

# instances/client-dupont/prod/topology.yaml
apiVersion: cac-configmgr.io/v1
kind: TopologyInstance
metadata:
  name: dupont-prod
  extends: acme-corp/profiles/enterprise
  fleetRef: ../fleet.yaml # ← Links to deployment target

spec:
  # Variables for interpolation
  vars:
    clientCode: DUPONT
    region: EU-WEST
    complianceLevel: high

  # Instance-specific overrides only
  repos:
    - name: repo-secu
      hiddenrepopath:
        - _id: fast-tier

```

```

        retention: 2      # ← Exception: even shorter retention
        # path inherited from chain
    - _id: warm-tier
        retention: 30    # ← Override
    - _id: nfs-tier
        retention: 2555  # ← 7 years instead of 10

```

Final API payload (after resolution and `_id` removal):

```
{
  "repos": [
    {
      "name": "repo-secu",
      "hiddenrepopath": [
        {"path": "/opt/immune/storage", "retention": 2},
        {"path": "/opt/immune/storage-warm", "retention": 30},
        {"path": "/opt/immune/storage-nfs", "retention": 2555}
      ]
    }
  ]
}
```

5. Resolution Algorithm

5.1 Inheritance Resolution

```

def resolve_template(instance):
    """
    Resolve full configuration from instance up to root.
    """

    # Build inheritance chain
    chain = []
    current = instance
    while current:
        chain.insert(0, current)  # Root first
        current = load_parent(current.extends)

    # Merge resources
    resolved = {}
    for template in chain:
        for resource_type, resources in template.spec.items():
            if resource_type == 'vars':
                # Variables merge shallow
                resolved['vars'] = merge_vars(
                    resolved.get('vars', {}),
                    resources
                )
            else:

```

```
# Resources merge by name
resolved[resource_type] = merge_resources(
    resolved.get(resource_type, []),
    resources
)

return resolved

def merge_resources(base_list, override_list):
    """
    Merge strategy: By resource name for top level, _id for list elements
    - Same name: Deep merge (recurse into lists with _id matching)
    - New name: Append
    - Explicit delete: Remove
    """
    result = {r['name']: r for r in base_list}

    for resource in override_list:
        name = resource['name']

        # Check for explicit deletion
        if resource.get('_action') == 'delete':
            result.pop(name, None)
            continue

        if name in result:
            # Deep merge: merge fields, with special handling for lists
            result[name] = deep_merge(result[name], resource)
        else:
            # New resource
            result[name] = resource

    return list(result.values())

def deep_merge(base, override):
    """
    Deep merge two dictionaries.
    For lists: match by _id, merge elements
    For dicts: recursive merge
    For primitives: override wins
    """
    result = dict(base) # Shallow copy

    for key, override_val in override.items():
        if key.startswith('_'):
            # Skip internal fields like _id
            continue

        if key not in result:
            # New field
            result[key] = override_val
        elif isinstance(override_val, list) and isinstance(result[key], list):
            # Merge lists by _id
            result[key] = deep_merge(result[key], override_val)
        else:
            result[key] = override_val

    return result
```

```
        result[key] = merge_list_by_id(result[key], override_val)
    elif isinstance(override_val, dict) and isinstance(result[key], dict):
        # Recursive merge for nested dicts
        result[key] = deep_merge(result[key], override_val)
    else:
        # Primitive override
        result[key] = override_val

    return result

def merge_list_by_id(base_list, override_list):
    """
    Merge two lists by matching _id fields.
    Elements without _id are treated as append-only.
    """
    # Index base list by _id
    base_by_id = {}
    base_without_id = []

    for item in base_list:
        if isinstance(item, dict) and '_id' in item:
            base_by_id[item['_id']] = item
        else:
            base_without_id.append(item)

    # Process overrides
    result = list(base_without_id) # Keep items without _id

    for override_item in override_list:
        if not isinstance(override_item, dict):
            # Primitive in list, append
            result.append(override_item)
            continue

        item_id = override_item.get('_id')

        if item_id and item_id in base_by_id:
            # Merge existing element
            base_elem = base_by_id[item_id]
            merged_elem = deep_merge(base_elem, override_item)

            # Check for deletion
            if override_item.get('_action') == 'delete':
                base_by_id.pop(item_id)
            else:
                base_by_id[item_id] = merged_elem
        else:
            # New element (no _id or new _id)
            if override_item.get('_action') != 'delete':
                if item_id:
                    base_by_id[item_id] = override_item
                else:
                    result.append(override_item)
```

```
# Add all merged elements with _id
result.extend(base_by_id.values())

return result
```

5.2 ID Filtering for API

After template resolution, **_id fields and _action fields must be removed** before sending to LogPoint API:

```
def filter_internal_ids(obj):
    """Remove all keys starting with _ from object."""
    if isinstance(obj, dict):
        return {
            k: filter_internal_ids(v)
            for k, v in obj.items()
            if not k.startswith('_')
        }
    elif isinstance(obj, list):
        return [filter_internal_ids(item) for item in obj]
    else:
        return obj

# Before API call
final_payload = filter_internal_ids(resolved_config)
```

Example transformation:

```
# Resolved template (internal)
repos:
  - name: repo-secu
    hiddenrepopath:
      - _id: fast-tier
        _action: merge
        path: /opt/immune/storage
        retention: 7
      - _id: warm-tier
        path: /opt/immune/storage-warm
        retention: 90

# After ID filtering (sent to API)
{
  "repos": [
    {
      "name": "repo-secu",
      "hiddenrepopath": [
        {"path": "/opt/immune/storage", "retention": 7},
        {"path": "/opt/immune/storage-warm", "retention": 90}
      ]
    }
  ]
}
```

```

    }
]
}

```

5.3 Variable Interpolation

After resolution, variables are interpolated:

```

# Template
spec:
  vars:
    basePath: /opt/immune/storage

  repos:
    - name: default
      hiddenrepopath:
        - path: "${vars.basePath}/default"
          retention: 90

# Instance
spec:
  vars:
    basePath: /mnt/custom/storage # ← Override

# Final resolved
repos:
  - name: default
    hiddenrepopath:
      - path: /mnt/custom/storage/default # ← Interpolated
        retention: 90

```

6. Deployment Profiles

6.1 SIMPLE Profile

Target: Single AIO, small log volume, basic compliance

```

# templates/profiles/simple.yaml
spec:
  repos:
    - name: default
      hiddenrepopath:
        - path: /opt/immune/storage
          retention: 90

  routingPolicies:
    - policy_name: rp-default
      catch_all: default

```

```

processingPolicies:
  - policy_name: pp-default
    norm_policy: auto
  
```

Characteristics:

- Single repo
- No rotation (single tier)
- Default policies only
- 5-10 configuration resources

6.2 MEDIUM Profile

Target: Distributed deployment, medium volume, standard retention

```

# templates/profiles/medium.yaml
spec:
  repos:
    - name: repo-secu
      hiddenrepopath:
        - path: /opt/immune/storage
          retention: 30
        - path: /opt/immune/storage-warm
          retention: 365

    - name: repo-system
      hiddenrepopath:
        - path: /opt/immune/storage
          retention: 90

    - name: repo-archive
      hiddenrepopath:
        - path: /opt/immune/storage-warm
          retention: 90
        - path: /opt/immune/storage-cold
          retention: 1095

  routingPolicies:
    - policy_name: rp-secu
      catch_all: repo-secu
      routing_criteria:
        - type: KeyPresent
          key: alert_severity

    - policy_name: rp-default
      catch_all: repo-system

  # ... more resources
  
```

Characteristics:

- 3-5 repos with rotation
- Tiered storage (fast → warm)
- Role-based routing
- 15-25 configuration resources

6.3 ENTERPRISE Profile

Target: Multi-cluster, high volume, strict compliance

```
# templates/profiles/enterprise.yaml
spec:
repos:
- name: repo-secu-critical
hiddenrepopath:
- path: /opt/immune/storage
retention: 7
- path: /opt/immune/storage-warm
retention: 90
- path: /opt/immune/storage-nfs
retention: 3650

- name: repo-secu-verbose
hiddenrepopath:
- path: /opt/immune/storage
retention: 7
- path: /opt/immune/storage-warm
retention: 30

- name: repo-system-critical
hiddenrepopath:
- path: /opt/immune/storage
retention: 30
- path: /opt/immune/storage-warm
retention: 365
- path: /opt/immune/storage-nfs
retention: 2555

- name: repo-system-verbose
hiddenrepopath:
- path: /opt/immune/storage
retention: 7
- path: /opt/immune/storage-warm
retention: 30

- name: repo-cloud
hiddenrepopath:
- path: /opt/immune/storage
retention: 180

- name: repo-archive-compliance
```

```

hiddenrepopath:
  - path: /opt/immune/storage-warm
    retention: 90
  - path: /opt/immune/storage-cold
    retention: 2555
  - path: /opt/immune/storage-nfs
    retention: 3650

- name: repo-trading-hft
hiddenrepopath:
  - path: /opt/immune/storage
    retention: 2
  - path: /opt/immune/storage-warm
    retention: 30

# Complex routing, enrichment, multiple processing policies
# ... 50+ configuration resources

```

Characteristics:

- 7+ repos with complex rotation (3-4 tiers)
- Fast → Warm → Cold → NFS archiving
- Granular routing (severity, source type, compliance flags)
- Advanced enrichment and processing
- 50+ configuration resources

7. File Organization

7.1 Directory Structure

```

cac-configmgr-templates/
└── logpoint/                                # Level 1: Golden Templates
    ├── golden-base.yaml
    ├── golden-pci-dss.yaml
    ├── golden-iso27001.yaml
    └── versions/
        ├── golden-base-v1.0.0.yaml
        ├── golden-base-v1.1.0.yaml
        └── golden-base-v2.0.0.yaml

└── mssp/                                     # Level 2: MSSP Templates
    └── acme-corp/
        ├── base.yaml                  # Extends logpoint/golden
        ├── acme-pci.yaml              # Extends base + PCI overlay
        └── profiles/
            ├── simple.yaml            # Extends acme/base
            ├── medium.yaml
            └── enterprise.yaml

└── instances/                                # Level 4: Concrete Instances

```

```

└── acme-corp/
    ├── client-dupont/
    │   ├── prod/
    │   │   ├── topology.yaml      # Extends enterprise profile
    │   │   ├── fleet.yaml
    │   │   └── vars.yaml         # Instance variables
    │   └── staging/
    │       ├── topology.yaml
    │       └── fleet.yaml
    └── client-martin/
        └── ...

```

7.2 Versioning Strategy

Templates are immutable and versioned:

```

metadata:
  name: golden-base
  version: "2.1.0" # SemVer

# References can pin version
extends: logpoint/golden-base@v2.1.0 # Exact
extends: logpoint/golden-base@v2       # Latest 2.x
extends: logpoint/golden-base          # Latest (risky)

```

8. Implementation Considerations

8.1 Performance

- **Resolution caching:** Cache resolved templates in `.cac-configmgr/cache/`
- **Delta calculation:** Only changed resources trigger API calls
- **Lazy loading:** Load templates on-demand during resolution

8.2 Validation

```

# Validate template hierarchy
cac-configmgr template validate --template templates/mssp/acme/base.yaml

# Check for conflicts
cac-configmgr template check-conflicts \
  --base logpoint/golden-base \
  --override mssp/acme/base

# Preview resolved configuration
cac-configmgr template resolve \
  --instance instances/client-dupont/prod/topology.yaml \
  --output resolved.yaml

```

8.3 Security

- **Template signing:** Golden templates signed by LogPoint
 - **Verification:** `cac-configmgr template verify --template <path>`
 - **Audit trail:** Track who changed which template when
-

9. Migration from DirSync

9.1 DirSync to CaC-ConfigMgr Mapping

DirSync Concept	CaC-ConfigMgr Equivalent
<code>base_config.yaml</code>	ConfigTemplate (intermediate level)
<code>inventory.yaml</code>	Fleet + TopologyInstance
<code>vars</code> in inventory	<code>spec.vars</code> in instance
Overlay files	Template inheritance chain
Jinja2 rendering	Variable interpolation post-resolution

9.2 Migration Path

1. Convert DirSync `base_config.yaml` to ConfigTemplate
 2. Extract inventory connection details to Fleet
 3. Extract variables to TopologyInstance
 4. Establish inheritance relationships
 5. Validate and test
-

10. Open Questions for Engineering

1. **Maximum inheritance depth:** Limit to prevent circular dependencies and stack overflow?
 - Proposal: Max 5 levels (LogPoint → MSSP → Profile → Instance → Environment)
2. **Conflict resolution:** When parent and child define same resource with incompatible fields?
 - Proposal: Child always wins (override)
3. **Partial override syntax:** Need for fine-grained patches (e.g., add one routing criteria without rewriting whole policy)?
 - Proposal: Use `action: patch` with JSON Patch-like syntax
4. **Template dependencies:** Can templates depend on other templates (not just extend)?
 - Example: Policy A depends on List B defined in another template
 - Proposal: Use `imports` for cross-template references

5. Real-time updates:

When Golden Template updates, how to propagate to running instances?

- Proposal: Drift detection alerts, manual reconciliation by default
-

Appendix A: Complete Example

See [examples/hierarchical-template/](#) for full working examples of:

- Simple profile deployment
- Enterprise multi-cluster configuration
- PCI-DSS compliance template inheritance

Appendix B: JSON Schema

See [schemas/template-v1.json](#) for complete JSON Schema validation.

Appendix C: Complete Repo Example with Template IDs

This example demonstrates the full inheritance chain using the **LogPoint Golden Template** with the 6 standard MSSP repos, progressively specialized by an MSSP and then by client type.

Step 1: LogPoint Golden Template

LogPoint provides the MSSP baseline with 6 standard repos + default. IMPORTANT: Only the OOB mount point `/opt/immune/storage` is used at this level.

```
# templates/logpoint/golden-mssp.yaml
apiVersion: cac-configmgr.io/v1
kind: ConfigTemplate
metadata:
  name: golden-mssp
  version: "1.0.0"
  provider: logpoint

spec:
  repos:
    # All repos use the SAME mount point /opt/immune/storage
    # LogPoint manages separation internally, not via filesystem paths

    - name: repo-default
      hiddenrepopath:
        - _id: primary
          path: /opt/immune/storage
          retention: 90

    - name: repo-secu
      hiddenrepopath:
        - _id: primary
          path: /opt/immune/storage
          retention: 365
```

```

- name: repo-secu-verbose
hiddenrepopath:
  - _id: primary
    path: /opt/immune/storage
    retention: 30

- name: repo-system
hiddenrepopath:
  - _id: primary
    path: /opt/immune/storage
    retention: 180

- name: repo-system-verbose
hiddenrepopath:
  - _id: primary
    path: /opt/immune/storage
    retention: 30

- name: repo-cloud
hiddenrepopath:
  - _id: primary
    path: /opt/immune/storage
    retention: 180

- name: repo-system-expert
hiddenrepopath:
  - _id: primary
    path: /opt/immune/storage
    retention: 730

```

IMPORTANT: At LogPoint level, ALL repos use the SAME OOB mount point `/opt/immune/storage`. The `path` field is ALWAYS the mount point path, never a subdirectory. LogPoint manages repo separation internally. Multi-tier rotation with different mount points (`storage-warm`, `storage-cold`, `storage-nfs`) is introduced by MSSPs.

Step 2: MSSP Base Template

ACME Corp (MSSP) inherits from LogPoint and:

- **Introduces new mount points:** `/opt/immune/storage-warm`, `/opt/immune/storage-cold`
- **Overrides** all retentions to their standard (more conservative)
- **Adds rotation tiers** using the new mount points for compliance-heavy repos
- **Keeps** the same 6 repos + default structure

```

# templates/mssp/acme-corp/base.yaml
apiVersion: cac-configmgr.io/v1
kind: ConfigTemplate
metadata:
  name: acme-base
  extends: logpoint/golden-mssp
  version: "1.0.0"

```

```
provider: acme-mssp

spec:
repos:
# DEFAULT: Just change retention
- name: repo-default
hiddenrepopath:
- _id: primary
  retention: 180 # Override: 90→180
  # path inherited: /opt/immune/storage

# SECU: Add warm tier (same path on different mount)
- name: repo-secu
hiddenrepopath:
- _id: primary
  retention: 90 # Override: 365→90 (shorter on fast storage)
- _id: warm
  path: /opt/immune/storage-warm # ← New mount point
  retention: 365 # ACME adds warm tier

# SECU-VERBOSE: Keep short, add warm
- name: repo-secu-verbose
hiddenrepopath:
- _id: primary
  retention: 7 # Override: 30→7 (very short on fast)
- _id: warm
  path: /opt/immune/storage-warm
  retention: 30 # Moved to warm

# SYSTEM: Add rotation
- name: repo-system
hiddenrepopath:
- _id: primary
  retention: 90 # Override: 180→90
- _id: warm
  path: /opt/immune/storage-warm
  retention: 180 # Moved to warm

# SYSTEM-VERBOSE: Keep simple
- name: repo-system-verbose
hiddenrepopath:
- _id: primary
  retention: 14 # Override: 30→14
  # No warm tier for verbose (low value)

# CLOUD: Standard rotation
- name: repo-cloud
hiddenrepopath:
- _id: primary
  retention: 90 # Override: 180→90
- _id: warm
  path: /opt/immune/storage-warm
  retention: 180
```

```
# EXPERT: Long retention with archive
- name: repo-system-expert
hiddenrepopath:
  - _id: primary
    retention: 90 # Override: 730→90
  - _id: warm
    path: /opt/immune/storage-warm
    retention: 365
  - _id: cold
    path: /opt/immune/storage-cold
    retention: 730 # ACME adds cold tier
```

Step 3: Profile by Deployment Type

ACME Corp creates an ENTERPRISE profile for large clients with strict compliance:

- **Introduces NFS mount point:** /opt/immune/storage-nfs (new at this level)
- **Adds NFS archive tier** for regulatory repos
- **Shortens fast-tier** even more (expensive storage)
- **Extends warm/cold** for audit requirements

```
# templates/mssp/acme-corp/profiles/enterprise.yaml
apiVersion: cac-configmgr.io/v1
kind: ConfigTemplate
metadata:
  name: acme-enterprise
  extends: acme-corp/base

spec:
  repos:
    # SECU: Add NFS for regulatory compliance (3 years)
    - name: repo-secu
      hiddenrepopath:
        - _id: primary
          retention: 7 # Override: 90→7 (very short, expensive)
        - _id: warm
          retention: 90 # Override: 365→90
        - _id: nfs
          path: /opt/immune/storage-nfs # ← New mount point
          retention: 1095 # Enterprise adds: 3 years regulatory

    # SECU-VERBOSE: Add archive
    - name: repo-secu-verbose
      hiddenrepopath:
        - _id: primary
          retention: 2 # Override: 7→2
        - _id: warm
          retention: 7 # Override: 30→7
        - _id: nfs
          path: /opt/immune/storage-nfs
          retention: 30 # Short archive
```

```

# SYSTEM: Extend for enterprise audit
- name: repo-system
hiddenrepopath:
  - _id: primary
    retention: 7      # Override: 90→7
  - _id: warm
    retention: 90     # Override: 180→90
  - _id: cold
    path: /opt/immune/storage-cold  # ← Same mount point as ACME
    retention: 730    # Enterprise adds: 2 years

# EXPERT: Maximum retention for forensics
- name: repo-system-expert
hiddenrepopath:
  - _id: primary
    retention: 7      # Override: 90→7
  - _id: warm
    retention: 90     # Override: 365→90
  - _id: cold
    retention: 365    # Override: 730→365
  - _id: nfs
    path: /opt/immune/storage-nfs
    retention: 2555   # Enterprise adds: 7 years legal hold

```

Step 4: Client Instance (Banque Dupont)

Specific client overrides for banking regulations:

- **Longer NFS retention** (10 years for banking law)
- **Shorter warm tiers** (faster compliance review)

```

# instances/banque-dupont/prod/topology.yaml
apiVersion: cac-configmgr.io/v1
kind: TopologyInstance
metadata:
  name: banque-dupont-prod
  extends: acme-corp/profiles/enterprise
  fleetRef: ./fleet.yaml

spec:
  vars:
    clientCode: DUPONT
    compliance: banking

repos:
  # SECU: Banking law requires 10 years
  - name: repo-secu
    hiddenrepopath:
      - _id: nfs
        retention: 3650  # Override: 1095→3650 (10 years)

```

```

# path inherited: /opt/immune/storage-nfs

# SECU-VERBOSE: Banking requires longer too
- name: repo-secu-verbose
hiddenrepopath:
- _id: nfs
  retention: 365 # Override: 30→365

# EXPERT: Legal hold extended
- name: repo-system-expert
hiddenrepopath:
- _id: nfs
  retention: 3650 # Override: 2555→3650 (10 years)

# NEW: Trading-specific repo (banking-specific)
- name: repo-trading
hiddenrepopath:
- _id: primary
  path: /opt/immune/storage
  retention: 1 # Very short (high frequency)
- _id: warm
  path: /opt/immune/storage-warm
  retention: 7
- _id: nfs
  path: /opt/immune/storage-nfs
  retention: 2555 # 7 years MiFID

```

Final Resolved Configuration for Banque Dupont

After inheritance resolution and _id filtering (all paths are mount points):

```

repos:
# 1. DEFAULT (single mount)
- name: repo-default
hiddenrepopath:
- path: /opt/immune/storage
  retention: 180

# 2. SECU (3 mount points)
- name: repo-secu
hiddenrepopath:
- path: /opt/immune/storage
  retention: 7
- path: /opt/immune/storage-warm
  retention: 90
- path: /opt/immune/storage-nfs
  retention: 3650

# 3. SECU-VERBOSE (3 mount points)
- name: repo-secu-verbose
hiddenrepopath:

```

```
- path: /opt/immune/storage
  retention: 2
- path: /opt/immune/storage-warm
  retention: 7
- path: /opt/immune/storage-nfs
  retention: 365

# 4. SYSTEM (3 mount points)
- name: repo-system
  hiddenrepopath:
    - path: /opt/immune/storage
      retention: 7
    - path: /opt/immune/storage-warm
      retention: 90
    - path: /opt/immune/storage-cold
      retention: 730

# 5. SYSTEM-VERBOSE (single mount)
- name: repo-system-verbose
  hiddenrepopath:
    - path: /opt/immune/storage
      retention: 14

# 6. CLOUD (2 mount points)
- name: repo-cloud
  hiddenrepopath:
    - path: /opt/immune/storage
      retention: 90
    - path: /opt/immune/storage-warm
      retention: 180

# 7. EXPERT (4 mount points)
- name: repo-system-expert
  hiddenrepopath:
    - path: /opt/immune/storage
      retention: 7
    - path: /opt/immune/storage-warm
      retention: 90
    - path: /opt/immune/storage-cold
      retention: 365
    - path: /opt/immune/storage-nfs
      retention: 3650

# 8. TRADING (Instance-specific, 3 mounts)
- name: repo-trading
  hiddenrepopath:
    - path: /opt/immune/storage
      retention: 1
    - path: /opt/immune/storage-warm
      retention: 7
    - path: /opt/immune/storage-nfs
      retention: 2555
```

Summary of Inheritance Chain

Key Principle: `path` is ALWAYS the mount point. Multiple repos can use the same path on different mount points for tiered rotation.

Repo	Tiers	Mount Points Used	Retention Evolution	Source of Final Values
default	1	/opt/immune/storage	90 → 180	ACME
secu	3	storage → storage-warm → storage-nfs	365 → 90 → 7 / 365 → 90 / 1095 → 3650	Enterprise(Instance)
secu-verbose	3	storage → storage-warm → storage-nfs	30 → 7 → 2 / 30 → 7 / 30 → 365	Instance
system	3	storage → storage-warm → storage-cold	180 → 90 → 7 / 180 → 90 / 730	Enterprise
system-verbose	1	/opt/immune/storage	30 → 14	ACME
cloud	2	storage → storage-warm	180 → 90 / 180	ACME
system-expert	4	storage → storage-warm → storage-cold → storage-nfs	730 → 90 → 7 / 365 → 90 / 730 → 365 / 2555 → 3650	Enterprise(Instance)
trading	3	storage → storage-warm → storage-nfs	Instance	Instance

Appendix D: Routing Policies Example

This example demonstrates how Routing Policies work with the hierarchical template system.

IMPORTANT: Routing Policies are **source-specific**, not generic. Each source (Windows, Linux, Checkpoint, Fortinet, etc.) has its own unique fields and criteria. There is no "generic firewall" policy that becomes "Checkpoint-specific" - each vendor has completely different field names and structures.

D.1 LogPoint Golden Template (Source-Specific)

LogPoint provides routing policies for **major source types**, each with their specific fields:

Important: Routing criteria have a `drop` field for filtering unwanted events:

- `drop: store` (default) - Store the log
- `drop: discard_raw` - Discard raw log but keep normalized
- `drop: discard` or `drop: discard_entirely` - Drop completely

Optimization: If a criterion routes to the same repo as `catch_all`, it is redundant and should be removed.

```
# templates/logpoint/golden-mssp/routing.yaml
apiVersion: cac-configmgr.io/v1
kind: ConfigTemplate
metadata:
  name: golden-mssp-routing
  version: "1.0.0"

spec:
  routingPolicies:
    # Windows - specific Windows Event fields
    - policy_name: rp-windows
      _id: rp-windows
      catch_all: repo-system
      routing_criteria:
        - _id: crit-verbose
          repo: repo-system-verbose
          type: KeyPresentValueMatches
          key: EventType
          value: Verbose
          drop: store
        - _id: crit-debug
          repo: repo-system-verbose
          type: KeyPresent
          key: DebugMode
          drop: store
        - _id: crit-drop-healthcheck # Drop unwanted events
          repo: repo-system-verbose
          type: KeyPresentValueMatches
          key: EventID
          value: "8000" # Health check events
          drop: discard_entirely

    # Linux - specific syslog fields
    - policy_name: rp-linux
      _id: rp-linux
      catch_all: repo-system
      routing_criteria:
        - _id: crit-debug
          repo: repo-system-verbose
          type: KeyPresentValueMatches
          key: severity
          value: debug
          drop: store
        - _id: crit-drop-cron # Drop cron spam
          type: KeyPresentValueMatches
          key: program
          value: "CRON"
          drop: discard_entirely
          # Note: no repo needed when dropping

    # Checkpoint - specific Checkpoint fields
    - policy_name: rp-checkpoint
      _id: rp-checkpoint
```

```
catch_all: repo-secu
routing_criteria:
  - _id: crit-detailed
    repo: repo-secu-verbose
    type: KeyPresentValueMatches
    key: fw_log_type
    value: detailed
    drop: store
  - _id: crit-smartview
    repo: repo-secu-verbose
    type: KeyPresent
    key: smartview_alert
    drop: store
  - _id: crit-drop-keepalive # Drop keepalives
    type: KeyPresentValueMatches
    key: msg_id
    value: "3000" # Keepalive messages
    drop: discard_raw

# Fortinet - specific Fortinet fields (different from Checkpoint!)
- policy_name: rp-fortinet
  _id: rp-fortinet
  catch_all: repo-secu
  routing_criteria:
    - _id: crit-information
      repo: repo-secu-verbose
      type: KeyPresentValueMatches
      key: loglevel
      value: information
      drop: store
    - _id: crit-utm
      repo: repo-secu
      type: KeyPresent
      key: utm_event
      drop: store

# Stormshield - specific Stormshield fields
- policy_name: rp-stormshield
  _id: rp-stormshield
  catch_all: repo-secu
  routing_criteria:
    - _id: crit-traffic
      repo: repo-secu-verbose
      type: KeyPresentValueMatches
      key: alarm_type
      value: traffic
      drop: store

# SentinelOne - EDR-specific fields
- policy_name: rp-sentinelone
  _id: rp-sentinelone
  catch_all: repo-system-expert
  routing_criteria:
    # Note: All SentinelOne logs go to expert by default (catch_all)
```

```
# No specific routing criteria needed unless filtering
- _id: crit-drop-heartbeat # Drop heartbeat noise
  type: KeyPresentValueMatches
  key: activityType
  value: "Heartbeat"
  drop: discard_entirely

# CrowdStrike - different EDR fields
- policy_name: rp-crowdstrike
  _id: rp-crowdstrike
  catch_all: repo-system-expert
  routing_criteria:
    # All CrowdStrike detections go to expert by default (catch_all)
    # Filter out informational events
    - _id: crit-drop-info
      type: KeyPresentValueMatches
      key: eventSeverity
      value: "Informational"
      drop: discard_entirely

# Darktrace - NDR-specific fields
- policy_name: rp-darktrace
  _id: rp-darktrace
  catch_all: repo-system-expert
  routing_criteria:
    # All Darktrace events go to expert by default
    # Filter low-confidence anomalies
    - _id: crit-drop-low-confidence
      type: KeyPresentValueMatches
      key: anomalyScore
      value: "low"
      drop: discard_entirely

# LogPoint NDR
- policy_name: rp-logpoint-ndr
  _id: rp-logpoint-ndr
  catch_all: repo-system-expert
  routing_criteria: []

# Office 365 - Cloud-specific fields
- policy_name: rp-o365
  _id: rp-o365
  catch_all: repo-cloud
  routing_criteria:
    # All 0365 logs go to cloud by default (catch_all)
    # Drop regular mailbox sync events
    - _id: crit-drop-sync
      type: KeyPresentValueMatches
      key: operation
      value: "SyncMailbox"
      drop: discard_entirely
```

D.2 MSSP Extension (Modifications)

MSSP inherits source-specific policies and modifies them. Each policy is independent - there is no inheritance between Windows and Linux or between Checkpoint and Fortinet.

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

spec:
  routingPolicies:
    # Windows: Modify retention, add PowerShell criterion
    - policy_name: rp-windows
      _id: rp-windows          # Same _id = merge
      catch_all: repo-system
      routing_criteria:
        - _id: crit-verbose
          repo: repo-system-verbose
          type: KeyPresentValueMatches
          key: EventType
          value: Verbose
          # No retention specified = inherited from parent
        - _id: crit-debug        # Inherited as-is
        - _id: crit-powershell   # New criterion added
          repo: repo-system-verbose
          type: KeyPresent
          key: PowershellCommand

    # Linux: Change debug threshold
    - policy_name: rp-linux
      _id: rp-linux
      catch_all: repo-system
      routing_criteria:
        - _id: crit-debug        # Override existing
          repo: repo-system-verbose
          type: KeyPresentValueMatches
          key: severity
          value: debug
          # Change from parent: different matching logic

    # Checkpoint: Keep as-is (no changes)
    # Since not redefined, fully inherited from LogPoint

    # Fortinet: Add DNS criteria
    - policy_name: rp-fortinet
      _id: rp-fortinet
      catch_all: repo-secu
      routing_criteria:
        - _id: crit-information  # Inherited
```

```

    - _id: crit-utm           # Inherited
    - _id: crit-dns           # New
      repo: repo-secu-verbose
      type: KeyPresent
      key: dns_query

# New source: Palo Alto (not in LogPoint template)
- policy_name: rp-paloalto
  _id: rp-paloalto          # New _id = new policy
  catch_all: repo-secu
  routing_criteria:
    - _id: crit-traffic
      repo: repo-secu-verbose
      type: KeyPresentValueMatches
      key: subtype
      value: wildfire

```

D.3 No Cross-Source Inheritance

Critical rule: Routing policies are NOT inherited across source types because each source has completely different field structures.

Source	Field for Verbose Logs	Example Value
Windows	EventType	Verbose
Linux	severity	debug
Checkpoint	fw_log_type	detailed
Fortinet	loglevel	information
SentinelOne	deepvisibility	(presence)

There is no "generic" field name that works across sources.

D.4 Merge Resolution Example

Windows Policy Resolution:

```

# LogPoint parent (rp-windows)
catch_all: repo-system
routing_criteria:
  - _id: crit-verbose
    repo: repo-system-verbose
    type: KeyPresentValueMatches
    key: EventType
    value: Verbose
  - _id: crit-debug
    repo: repo-system-verbose
    type: KeyPresent
    key: DebugMode

```

```

# MSSP child (rp-windows with _id: rp-windows)
catch_all: repo-system
routing_criteria:
  - _id: crit-verbose           # Same _id = override fields specified
    repo: repo-system-verbose
    type: KeyPresentValueMatches
    key: EventType
    value: Verbose
    # All other fields inherited (repo, type, etc.)
  - _id: crit-debug            # Not redefined = inherit as-is
  - _id: crit-powershell       # New _id = append
    repo: repo-system-verbose
    type: KeyPresent
    key: PowershellCommand

# Final resolved for Windows
catch_all: repo-system
routing_criteria:
  - _id: crit-verbose
    repo: repo-system-verbose
    type: KeyPresentValueMatches
    key: EventType
    value: Verbose
  - _id: crit-debug           # From parent (inherited)
    repo: repo-system-verbose
    type: KeyPresent
    key: DebugMode
  - _id: crit-powershell      # From child (added)
    repo: repo-system-verbose
    type: KeyPresent
    key: PowershellCommand

```

D.5 Source-to-Policy Mapping

Mapping links log sources to their routing policies:

```

# templates/mssp/acme-corp/source-mappings.yaml
spec:
  sourceMappings:
    # OS
    - vendor: microsoft
      product: windows
      routingPolicy: rp-windows

    - vendor: linux
      product: syslog
      routingPolicy: rp-linux

    # Firewalls
    - vendor: checkpoint

```

```

product: firewall
routingPolicy: rp-checkpoint

- vendor: fortinet
  product: fortiGate
  routingPolicy: rp-fortinet

- vendor: stormshield
  product: sns
  routingPolicy: rp-stormshield

- vendor: paloalto
  product: panos
  routingPolicy: rp-paloalto      # MSSP-added

# EDR/XDR
- vendor: sentinelone
  product: edr
  routingPolicy: rp-sentinelone

- vendor: crowdstrike
  product: falcon
  routingPolicy: rp-crowdstrike

# NDR
- vendor: darktrace
  product: immune-system
  routingPolicy: rp-darktrace

- vendor: logpoint
  product: ndr
  routingPolicy: rp-logpoint-ndr

# Cloud
- vendor: microsoft
  product: o365
  routingPolicy: rp-o365

```

D.6 Validation Rules for Routing Policies

1. Repository References:

All **repo** values MUST reference existing repositories.

- **Validation:** FAIL if referenced repo does not exist
- Example: **rp-windows** references **repo-system-verbose** → must exist in repos
- **Exception:** When **drop: discard*** is set, **repo** is optional (nothing to store)

2. Redundant Criteria:

A criterion that routes to the same repo as **catch_all** is redundant and should be removed or changed.

```
# BAD: crit-normal routes to same repo as catch_all
catch_all: repo-system
```

```

routing_criteria:
  - repo: repo-system # ← Redundant! Same as catch_all
    type: KeyPresent
    key: normal_log

# GOOD: Only route DIFFERENT repos or DROP
catch_all: repo-system
routing_criteria:
  - repo: repo-system-verbose # ← Different repo = useful
    type: KeyPresentValueMatches
    key: severity
    value: high
  - drop: discard_entirely # ← Drop spam = useful
    type: KeyPresent
    key: healthcheck

```

3. Drop Field: The **drop** field controls event filtering:

- **drop: store** (default) - Store in specified **repo**
- **drop: discard_raw** - Discard raw event
- **drop: discard** or **drop: discard_entirely** - Drop completely
- When dropping, **repo** field is optional

4. Source-Specific: Each routing policy is completely independent.

- No inheritance between **rp-checkpoint** and **rp-fortinet**
- No inheritance between **rp-windows** and **rp-linux**
- Merge only happens WITHIN the same **_id** (same source type across template levels)

5. Inheritance by _id (within same source):

- Same **_id** in parent and child = **merge** (child overrides specified fields)
- Missing **_id** in child but present in parent = **inherit** (copy as-is)
- New **_id** in child = **append** (add to end of criteria list)
- **_action: delete** on existing **_id** = **remove** criterion

6. New Sources: MSSP can add entirely new routing policies for sources not in LogPoint template (e.g., **rp-paloalto**).

7. Policy Uniqueness: **policy_name** must be unique within a template level.

D.7 Deletion Example

MSSP removes a criterion from Windows policy:

```

# Child template removes crit-debug from Windows
- policy_name: rp-windows
  _id: rp-windows
  catch_all: repo-system
  routing_criteria:
    - _id: crit-verbose

```

```

- _id: crit-debug
  _action: delete          # Removes this criterion
- _id: crit-powershell

```

D.8 Instance Override Examples

Example 1: Override catch_all

A forensic investigation requires **all** Windows logs to be stored in verbose repo:

LogPoint template:

```

- policy_name: rp-windows
  catch_all: repo-system

```

Instance override:

```

# instances/forensic-client/prod/routing-override.yaml
spec:
  routingPolicies:
    - policy_name: rp-windows
      _id: rp-windows
      catch_all: repo-system-verbose  # ← Override: all logs go to verbose
repo
      routing_criteria: [] # No filtering needed during investigation

```

Example 2: Add new filtering criteria

A bank adds compliance-specific filtering:

```

# instances/banque-dupont/prod/routing-override.yaml
spec:
  routingPolicies:
    - policy_name: rp-windows
      _id: rp-windows
      catch_all: repo-system
      routing_criteria:
        - _id: crit-verbose
          repo: repo-system-verbose
          type: KeyPresentValueMatches
          key: EventType
          value: Verbose
        - _id: crit-debug
          type: KeyPresent
          key: DebugMode
        - _id: crit-pii # ← New: filter PII events for GDPR
          type: KeyPresent

```

```
key: containsPII
drop: discard_entirely
key: EventType
value: Verbose
# Override: different repo or retention logic
```