**Task 20: Building and Configuring the Service Catalog**

**What It Is**

The Service Catalog is the front door of your service organization. It transforms vague requests and scattered emails into structured, traceable, and automated workflows. This chapter combines two essential phases: gathering service catalog requirements and translating those into live, usable catalog items in Freshservice.

Each item in the catalog represents a service a business unit provides—whether it's IT issuing hardware, HR processing PTO requests, or Facilities coordinating badge access. Through carefully defined fields, workflows, visibility rules, and automation, the catalog becomes a self-service engine that delivers clarity, efficiency, and value.

**Why It Matters**

An intelligently designed catalog:

* Reduces ticket errors and misroutes
* Promotes standardization
* Accelerates request fulfillment
* Encourages self-service and SLA tracking
* Supports automation and visibility

From an ITIL perspective, this task underpins:

* **Service Catalog Management** – defining and maintaining the services available to users
* **Request Fulfillment** – converting catalog items into repeatable workflows
* **Access Management** – granting controlled access to systems or resources
* **Service Level Management** – aligning requests to SLA expectations
* **Change Enablement** – managing the change impact of catalog-driven provisioning

Done well, the catalog becomes a dynamic, user-centered platform for service delivery.

**How To Do It**

**1. Gather Candidate Service Items**

* Request a list of services from stakeholders across IT, HR, and Facilities.
* Ask each team to describe their current workflows and legacy request methods.
* Limit the first wave to **25–30 high-value items** to avoid overloading the launch.

**2. Define Field Requirements for Each Item**

* Identify form fields such as:
  + Text fields (e.g., Justification)
  + Dropdowns (e.g., Department, Manager)
  + Attachments (e.g., Policy Forms)
  + Date pickers, checkboxes, or multi-select fields
* Note whether fields are:
  + Required or optional
  + Conditional (e.g., only show “Software Type” if “Software Request” is selected)

**3. Design for Visibility and Permissions**

* Use **group visibility rules** to restrict access:
  + Example: Only managers can submit "New Hire Requests"
* Limit catalog item visibility by:
  + Role (e.g., Agents, Requesters)
  + Department (e.g., Finance, HR)
  + Location (e.g., Remote vs. On-Site)

**4. Create Logical Categories**

* Group catalog items into no more than **10 core categories**:
  + Hardware (laptops, monitors)
  + Software (application installs)
  + Access (VPN, shared drives)
  + HR Services (PTO, onboarding)
  + Facilities (office supplies, badge reissue)
* Avoid siloed department naming unless it adds user clarity.

**5. Build the Catalog Items**

For each service item:

* **Name** – Use clear, action-based naming (e.g., “Request New Laptop”)
* **Description** – Explain what it’s for and when it should be used
* **Form Fields** – Add all defined and conditional fields
* **Approval Workflow** – Define approvers (e.g., direct manager, department head)
* **Fulfillment Group** – Assign the team responsible (e.g., IT Procurement)
* **Icon** – Choose a visual for scannability
* **Visibility** – Apply rules as defined earlier

**6. Plan and Embed Approval Logic**

* Document when approvals are needed:
  + Example: “If cost > $500, route to Department Head”
  + Use manager lookups or department-based triggers
* Set up approvals either:
  + Inline within the catalog item
  + Through **Workflow Automator** for complex logic

**7. Link Knowledge Base Articles**

* Pair catalog items with relevant solution articles
* Example: “Request VPN Access” → link to “How to Connect to the VPN”
* Help users self-educate before and after request submission

**8. Build Workflow Automators**

* For each major item, design automations such as:
  + **Assign tasks** (e.g., Device Prep for laptop)
  + **Send notifications** (e.g., HR confirmation)
  + **Escalate SLA breaches**
  + **Close tickets when all tasks complete**

**9. Test Before Publishing**

* Use the impersonate requester tool to validate the portal view
* Submit test requests for:
  + Field behavior
  + Approval routing
  + Fulfillment assignment
  + Notifications
* Fix any misconfiguration before go-live

**10. Launch and Promote**

* **Soft launch** with internal testing, then release in phases
* Use **banners or internal emails** to announce high-impact items
* Train teams to use the catalog instead of ad hoc emails

**Tips for Success**

* Start small and scale: focus on the most frequently requested items first
* Use friendly labels: “Need a Laptop?” is better than “Workstation Provisioning”
* Document ownership: Who updates each form and handles fulfillment?
* Monitor usage in Analytics: retire low-traffic items, tune popular ones
* Review items quarterly to align with policy, staffing, and tech changes

**ITIL Tie-In**

This task is central to several ITIL practices and is foundational to delivering standardized, traceable, and user-friendly services across the organization:

* **Service Catalog Management**  
  The entire task is built on this practice—defining, structuring, and publishing services in a way that is accessible and meaningful to users. It ensures users only see and request what’s appropriate for their role, department, and location.
* **Request Fulfillment**  
  Every catalog item becomes a repeatable, reportable workflow that reduces effort, ensures completeness, and promotes faster service delivery.
* **Access Management**  
  For service items that grant access (e.g., VPN, shared drives, systems), this task enforces visibility and approval controls, helping manage risk and compliance.
* **Service Level Management**  
  Catalog items support SLA alignment by defining expected fulfillment timelines, escalation rules, and visibility for stakeholders.
* **Change Enablement**  
  Some catalog items initiate changes to infrastructure or user access. This process allows automation to tie into CAB review or change templates as needed.
* **Knowledge Management**  
  Pairing catalog items with solution articles helps users make better decisions, increases self-service success, and builds a user-aware knowledge ecosystem.
* **Service Design**  
  By involving stakeholders early in the catalog definition and field requirement process, this task ensures services are usable, valuable, and aligned with business needs.

This task also reflects several **ITIL 4 guiding principles**, including:

* **Focus on Value** – Every service item must address a real user need
* **Keep it Simple and Practical** – Start with high-volume, low-friction services
* **Design for Collaboration** – Catalog success depends on clear input from each business unit
* **Progress Iteratively with Feedback** – Launch in phases and adjust based on usage metrics and user input

In short, your Service Catalog is more than a list of services—it’s a dynamic platform for enabling efficient, controlled, and value-aligned service delivery.