Understanding Playbooks

1. Playbook Basics

1.1 What Are Playbooks?

Playbooks are reusable workflow templates that define how AI analysis is applied to documents within Jylo. They contain specific filters and logic that determine what information to extract, how to process it, and how to present the results.

In Jylo 1.0, these were called "Products," but Playbooks in Jylo 2.0 provide enhanced capabilities with greater flexibility and power.

1.2 Benefits of Using Playbooks

Playbooks deliver several key advantages for your document analysis workflow:

Consistency: Apply the same analytical approach across all documents

Efficiency: Save time by reusing established analysis workflows

Customisation: Configure complex analyses with conditional logic

Quality Control: Build verification directly into the workflow

Collaboration: Share expertise through standardised processes

2. Working with Playbooks

2.1 Playbooks and Flows

It's important to understand how Playbooks fit into the Jylo workflow:

- **Playbooks** define the analysis pattern (what questions to ask, what logic to apply)
- **Flows** are where you apply these Playbooks to actual documents

You'll select which Playbook to use when you create a new Flow within a Project. This is when you'll decide which analytical approach is right for your specific documents.

2.2 The Playbook Marketplace

The Marketplace serves as a central repository where you can browse, access, and (with appropriate permissions) manage Playbooks.

To access the Marketplace:

- Navigate to **Marketplace** in the left sidebar
- Select **Playbooks** from the submenu

Each Playbook appears as a card displaying basic information. Clicking on a card shows you the overview with filter labels and descriptions.

Note: The Marketplace is particularly useful for users with Builder access who create and manage Playbooks. Regular users will primarily interact with Playbooks when creating Flows within Projects although all users are encouraged to browse the marketplace to see what's possible.

2.3 Playbook Access Control

Playbooks have specific access controls that determine who can use and modify them:

- **Builder permission:** Enables a user to clone, edit and modify the Playbook after publication
- **Consumer permission:** Makes the Playbook visible to users who can deploy it in Flows

Your access to specific Playbooks depends on the permissions assigned to you by the Playbook creator.

3. Playbook Components

A Playbook contains several key components:

- **Filters:** Questions that extract specific information from documents
- **Sub-filters:** Secondary questions able to build upon parent filter responses using hash-linking

- **Conditional Logic:** Rules determining when specific filters should be applied

3.1 Filter Types

Filters are the foundation of any Playbook and come in several types:

- Filtering-Capable Types (can analyse content AND filter documents):
  - **Yes/No:** For binary decisions (presented as checkboxes)
  - **List of values:** For multiple categorical items (presented as checkboxes)
  - **Range of values:** For numerical results (presented as a slider control)
- Analysis-Only Types (provide analysis but cannot filter documents):
  - **Text:** For a short string response with evidence
  - **Generative:** For long-form outputs without citation links

4. Advanced Playbook Features

4.1 Hash-Linking System

The hash-linking system enables sophisticated multi-stage analysis by connecting outputs from different filters:

- Use the # symbol to reference outputs from other filter questions
- Use the * symbol to reference input documents
- Use the @ symbol to reference source documents

For example, you may carry out a general assessment using a parent filter and expand on the analysis using a sub:

Build upon the general assessment by identifying risks associated with outstanding debt from the financial agreement.
*Financial Agreement: *inputdocument*
General Assessment: #filter01

Note: AI sees the entire contents of the reference symbol where you place it so keep it separate from instructions to ensure they are not distorted.

This powerful feature allows filters to build upon previous results, creating a chain of reasoning.

4.2 Conditional Logic

Conditional logic allows Playbooks to adapt their analysis based on document content or previous filter results:

- **Yes/No Filters**: Trigger actions based on yes/no responses
- **List of Values Filters**: Activate based on specific values in a list
- **Range of Values Filters**: Trigger within defined numerical ranges
- **Text Filters**: Activate when specific text patterns are found

Conditions can be applied multi-directionally across any filters regardless of hierarchy, creating sophisticated analysis pathways.

Note: Parent filters must be triggered for their sub-filters to be applied.

5. Creating Playbooks

You can create custom Playbooks from the Marketplace.

1. Navigate to **Marketplace > Playbooks**
2. Click the **blue plus button** in the top-right corner

Metadata

- Enter the Playbook Name and Description

Filters

- Set the Label, Description, Question, Answer Type, and Model
- Upload test and source documents using the right panel

Sub-filters and Conditional Logic

- **Add Condition** at the bottom of any filter

- Select a trigger filter and configure the condition parameters

Testing

- Create a test Flow with the blue plus button

Access Control

- Assign **Builder** or **Consumer** permissions to specific users

Summary

- Review your Playbook overview

Tip: When creating list of value filters ensure you have instructed the model as to the exact values it is allowed to return as these values are what you will use when configuring your conditions.

6. Next Steps

Now that you understand how Playbooks work, you're ready to apply them to your documents by creating Flows within a Project. Flows allow you to execute Playbooks against specific document sets and review the results.

To learn how to create and manage Flows, please refer to the Using Flows guide.

Quick Reference: Playbook Terms

Term	Definition
Playbook	Reusable workflow template for document analysis (formerly called "Products" in Jylo 1.0)
Filter	Foundation component that extracts specific information from documents
Sub-filter	Process component able to build upon parent filter responses
Conditional Logic	Rules that trigger filters based on other filter responses
Hash-Linking	Use # to reference filter output it other filters