

Project SPINDLE – RPG Publisher Silo Branch

Extension Purpose: Adapt SPINDLE's ingestion and enrichment system to handle RPG book PDFs and metadata extraction to build publisher silos and generate product-ready entries for your webstore.

Step 6: RPG Silo Extraction Mode

Add the following to `run_spindle.py`:

```
from RPGPDFIngestor import process_rpg_archive

if __name__ == '__main__':
    # ...existing code...
    process_rpg_archive("/mnt/data/RPG_LIBRARY/")
```

Create a new file `RPGPDFIngestor.py`:

```
import os
from pdfminer.high_level import extract_text
from pathlib import Path

PUBLISHER_PATH = Path("~/Vault/Publishers/").expanduser()
PUBLISHER_PATH.mkdir(parents=True, exist_ok=True)

KEYWORDS = ["dungeon", "pathfinder", "starfinder", "5e", "adventure",
            "campaign", "module"]

def process_rpg_archive(directory):
    for file in Path(directory).rglob("*.pdf"):
        try:
            text = extract_text(file)
            match = any(k in text.lower() for k in KEYWORDS)
            if match:
                name = file.stem.replace("_", " ").title()
                pub_dir = PUBLISHER_PATH / name[:1].upper() / name
                pub_dir.mkdir(parents=True, exist_ok=True)
                with open(pub_dir / "manifest.md", "w") as f:
                    f.write(f"# {name}\n\n**Source:** {file.name}\n\n##
Detected Keywords:\n")
                    for kw in KEYWORDS:
                        if kw in text.lower():
                            f.write(f"- {kw}\n")
                print(f"[RPG-SPINDLE] Archived: {file.name} → {pub_dir}")
            except Exception as e:
                print(f"Failed to process {file}: {e}")
```

Step 7: Product Card Drafting

Create a new script: `ProductCardGenerator.py`

```
import json, os
from pathlib import Path

MANIFESTS = Path("~/Vault/Publishers/").expanduser()
OUTPUT = Path("~/Vault/WebstoreDrafts/")
OUTPUT.mkdir(parents=True, exist_ok=True)

for manifest in MANIFESTS.rglob("manifest.md"):
    try:
        with open(manifest, "r") as f:
            lines = f.readlines()
            title = lines[0].strip("# ").strip()
            keywords = [l.strip("- ").strip() for l in lines if l.startswith("- ")]

        product = {
            "title": title,
            "tags": keywords,
            "source_file": manifest.parent.name,
            "drafted_by": "SPINDLE",
        }

        with open(OUTPUT / f"{title.replace(' ', '_')}.json", "w") as out:
            json.dump(product, out, indent=2)
    except Exception as e:
        print(f"Error drafting product for {manifest}: {e}")
```


Step 8: Codex Tagging and Publisher Agent Sync

Use:

```
@codex index('~/.Vault/Publishers/')
@codex learn_from('ProductCardGenerator', { format: 'json', target:
'shopify_drafts' })
```

Tag any new publishers with:

```
codex.tag_publisher("Frog God Games", ['osr', 'hardcover', 'hex crawl'])
```

 Once deployed, this branch allows agents to:

- Parse hundreds of RPG PDFs

- Organize and silo by publisher + keywords
- Auto-generate product listings for eCommerce
- Feed scroll hooks with publisher-specific logic (e.g. `Necrotic Gnome Rituals`)

Can be activated via:

```
python run_spindle.py --mode=rpg
```