

# Build faster with Solution Accelerators

## Foundry IQ + Fabric IQ

### Hands-on Workshop

Build AI applications that combine unstructured document knowledge with structured enterprise data using knowledge bases, ontology, and natural language queries.

*Generated: February 02, 2026*

# Table of Contents

---

<b>1. Introduction</b>	What you will build
<b>2. Prerequisites</b>	What you need before starting
<b>3. Architecture Overview</b>	Understanding the components
<b>4. Part 1: Run with Sample Scenario</b>	Retail example walkthrough
<b>5. Part 2: Customize Your Own Scenario</b>	Create your industry solution
<b>6. Understanding the Build Process</b>	What each step does
<b>7. Testing Your Agent</b>	Interactive chat with your data
<b>8. Troubleshooting</b>	Common issues and solutions

# 1. Introduction

---

Welcome to the **Foundry IQ + Fabric IQ** hands-on workshop! In this workshop, you will build an intelligent agent that combines unstructured document knowledge with structured enterprise data.

## The Opportunity

Organizations have valuable knowledge spread across documents (PDFs, policies, manuals) and structured systems (databases, data warehouses). By connecting these sources through AI, users can get unified answers from a single conversational interface.

## The Solution

**Foundry IQ** and **Fabric IQ** solve this by enabling an intelligent agent that:

- Creates knowledge bases from documents with agentic retrieval (plan, iterate, reflect)
- Defines business ontology to understand entities, relationships, and rules
- Queries data using natural language over both documents and structured data
- Combines both to answer complex business questions

## What You Will Build

By the end of this workshop, you will have created:

- **AI Agent** - Azure AI Foundry orchestrates tools and generates responses
- **Knowledge Base** - **Foundry IQ** provides agentic retrieval over documents
- **Business Ontology** - **Fabric IQ** defines entities, relationships, and NL-to-SQL
- A multi-tool agent that queries both structured and unstructured data

## Workshop Flow

This workshop is divided into two parts:

1. **Run with Sample Scenario** - Follow along with a pre-defined Retail scenario to understand how everything works.
2. **Customize Your Own** - Use AI to generate data and documents for YOUR industry and use case.

## 2. Prerequisites

---

### Required Tools

- Python 3.10 or higher
- Azure CLI (logged in with 'az login')
- Azure Developer CLI (azd)
- Git

### Required Azure Resources

The following resources will be created by the infrastructure deployment:

- Azure AI Services (GPT-4o-mini, text-embedding-ada-002)
- Azure AI Search (for document search)
- Azure Storage Account

### Required Fabric Resources

- Microsoft Fabric workspace with capacity assigned
- Fabric workspace ID (from the URL when viewing your workspace)

### Initial Setup

Before starting the workshop, run these commands:

```
# Clone the repository
git clone <repo-url>
cd fabric-ontology-lab

# Deploy Azure infrastructure
azd up

# Install Python dependencies
pip install -r scripts/requirements.txt
```

Then update your .env file with your Fabric workspace ID:

```
FABRIC_WORKSPACE_ID=your-workspace-id-here
```

## 3. Architecture Overview

---

The solution combines **Foundry IQ** and **Fabric IQ** to create an intelligent agent that can answer questions using both structured data (tables) and unstructured data (documents).

### Components

#### Foundry IQ (Document Intelligence)

- **Knowledge Base:** Agentic retrieval over documents (plan, iterate, reflect)
- **AI Search Index:** Vectorized document chunks for semantic search
- **Embedding Model:** Converts text to vectors (text-embedding-ada-002)

#### Fabric IQ (Structured Data Intelligence)

- **Fabric Lakehouse:** Stores structured data as Delta tables
- **Business Ontology:** Defines entities, properties, and relationships
- **Data Agent:** Translates natural language to SQL queries

#### Orchestration Layer

- **Azure AI Foundry Agent:** Orchestrates tools and generates responses
- **Multi-tool routing:** Determines which source(s) to query

### Data Flow

1. User asks a question in natural language
2. AI agent determines if the question needs structured data, documents, or both
3. For structured data: **Fabric IQ** converts question to SQL via Ontology
4. For documents: **Foundry IQ** retrieves relevant document chunks
5. AI agent combines results and generates a natural language response

## 4. Part 1: Run with Sample Scenario

---

In this section, you will run the complete build process with a pre-defined Retail scenario. This helps you understand how each component works before creating your own custom scenario.

### The Retail Scenario

Our sample scenario is a Retail system for inventory and sales tracking. It includes:

- Products table: Product catalog with prices and stock levels
- Transactions table: Sales transactions with quantities and dates
- Policy documents: Store policies and procedures (PDFs)

### Step 1: Run the Build Script

The build script automates all steps. Run this single command:

```
python scripts/00_build_solution.py \  
  --industry "Retail" \  
  --usecase "Inventory and sales tracking"
```

#### Note

This uses pre-generated sample data from the data/ folder. No AI generation is needed for the sample scenario.

### What the Build Script Does

1. Loads sample data (patients.csv, appointments.csv)
2. Creates Fabric Lakehouse and uploads data as Delta tables
3. Creates Fabric Ontology with entities and relationships
4. Generates PDF documents from the data
5. Uploads documents to AI Search with embeddings
6. Creates the Fabric Data Agent
7. Creates the Foundry orchestration agent

### Step 2: Test the Agent

Once the build completes, test your agent:

```
python scripts/08a_test_multi_tool_agent.py
```

Try these sample questions:

- How many products do we have?
- Show me the top 5 products by sales
- What is our return policy?
- Which products are low in stock?

## 5. Part 2: Customize Your Own Scenario

Now that you understand how the build process works, it is time to create your own custom scenario! The AI will generate realistic sample data and documents based on your industry and use case.

### Step 1: Choose Your Industry and Use Case

Think about what industry and use case you want to explore. Here are some ideas:

Industry	Use Case	Tables
Healthcare	Patient records and scheduling	patients, appointments
Finance	Loan applications and approvals	customers, loans
Education	Student enrollment and grades	students, courses
Manufacturing	Equipment maintenance	machines, repairs
Hospitality	Hotel reservations	guests, bookings
Real Estate	Property listings	properties, agents

### Step 2: Run with AI Generation

Add the --ai flag to generate custom data:

```
python scripts/00_build_solution.py --ai \
  --industry "YOUR_INDUSTRY" \
  --usecase "YOUR_USE_CASE"
```

For example, to create a Healthcare patient system:

```
python scripts/00_build_solution.py --ai \
  --industry "Healthcare" \
  --usecase "Patient records and appointment scheduling"
```

#### Tip

Be descriptive in your use case! The more detail you provide, the better the AI can generate relevant data and documents.

### Step 3: Switch Between Scenarios



When you want to try a different scenario, use the `--clean` flag to create fresh Fabric artifacts:

```
python scripts/00_build_solution.py --ai --clean \  
  --industry "Finance" \  
  --usecase "Loan applications and credit scoring"
```

The `--clean` flag increments the artifact suffix (lakehouse\_1 -> lakehouse\_2) so you can have multiple scenarios without conflicts.

## 6. Understanding the Build Process

---

The build process consists of several scripts that work together. Understanding each step helps you troubleshoot issues and customize behavior.

### **00\_build\_solution.py**

#### *Master orchestration script*

Runs all other scripts in sequence. Accepts --ai, --clean, --industry, and --usecase flags.

### **01a\_generate\_sample\_data.py**

#### *AI data generation*

Uses GPT-4o-mini to generate realistic CSV data and PDF documents based on your scenario.

### **02\_setup\_fabric.py**

#### *Fabric Lakehouse and Ontology*

Creates the Lakehouse for data storage and Ontology for semantic understanding.

### **03\_load\_fabric\_data.py**

#### *Data loading*

Uploads CSV files to OneLake and loads them as Delta tables.

### **04\_generate\_prompt.py**

#### *Schema extraction*

Reads table schemas to generate prompts for the AI agent.

### **05\_create\_fabric\_agent.py**

#### *Fabric Data Agent*

Creates a Data Agent in Fabric that uses the Ontology to answer questions.

### **06\_upload\_to\_search.py**

#### *Document indexing*

Uploads PDF documents to AI Search with vector embeddings.

### **07a\_create\_foundry\_agent.py**

#### *Foundry orchestration agent*

Creates the main AI agent that combines Fabric and Search tools.

## 7. Testing Your Agent

---

The test script provides an interactive chat interface to your AI agent. It shows you exactly how the agent processes your questions.

### Running the Test

```
python scripts/08a_test_multi_tool_agent.py
```

### Understanding the Output

When you ask a question, the agent shows:

- Tool calls: Which tools (Fabric IQ, Foundry IQ) were used
- SQL queries: The exact SQL generated by Fabric IQ
- Search results: Document chunks retrieved by Foundry IQ
- Final answer: The natural language response

### Sample Questions by Type

#### Structured Data Questions (uses Fabric IQ)

- How many records are in the table?
- Show me the top 5 items by value
- What is the average/sum/count of X?
- List all items where condition is met

#### Unstructured Data Questions (uses Foundry IQ)

- What is our policy on X?
- Tell me about the procedures for Y
- What guidelines exist for Z?

#### Combined Questions (uses both)

- Which customers have issues and what is the resolution policy?
- Show me overdue items and explain the escalation process

## 8. Troubleshooting

---

### Ontology stuck on 'Setting up'

The Ontology creation can sometimes get stuck. Use --clean flag to create a new one with an incremented suffix.

```
python scripts/00_build_solution.py --clean ...
```

### FABRIC\_WORKSPACE\_ID not set

Make sure your .env file contains the Fabric workspace ID. Get it from the Fabric portal URL.

```
FABRIC_WORKSPACE_ID=fb695e19-2010-...
```

### AI generation produces invalid data

The AI retry mechanism will attempt 3 times. If it keeps failing, try simplifying your use case description.

```
# Use simpler, more specific use case descriptions
```

### Rate limiting (429 errors)

The scripts have built-in retry logic. If you see many 429 errors, wait a few minutes and try again.

```
# Automatic retry with backoff is enabled
```

### Search returns no results

Make sure step 06 completed successfully. Check that PDFs were generated and uploaded.

```
python scripts/06_upload_to_search.py --data-folder <PATH>
```

### Fabric Data Agent not responding

The Data Agent needs time to index the Ontology. Wait 2-3 minutes after creation before testing.

```
# Wait for 'Agent is ready' message
```

## Congratulations!

You have successfully completed the Foundry IQ + Fabric IQ hands-on workshop!

You now know how to:

- Use Fabric IQ to create business ontologies and query structured data
  - Use Foundry IQ to build knowledge bases over documents
  - Build multi-tool AI agents with Azure AI Foundry
- Combine structured and unstructured data in intelligent queries

Keep experimenting with different industries and use cases!

*For questions and feedback, visit the [GitHub repository](#).*