# Method of Procedure (MOP)

## Integration of Azure DevOps Repository with Azure Databricks

### 1. Purpose

To establish and configure a connection between Azure DevOps (ADO) and Azure Databricks (ADB) for efficient source control and collaborative notebook development.

### 2. Scope

This procedure applies to the setup and synchronization of Databricks notebooks with an Azure DevOps Git repository within the Azure environment.

### 3. Prerequisites

* Access to Azure Portal with permissions to create and manage resources.
* A project created in Azure DevOps.
* Access to Azure Databricks workspace.
* Necessary permissions to create repositories and branches in ADO.
* Personal Access Token (PAT) in ADO (if authentication is required).

### 4. Procedure

#### Step 1: Azure DevOps Setup

1. Log in to the Azure DevOps portal.
2. Navigate to the Project created for the development work.
3. Create a new repository within this project to store Databricks notebooks.
4. Copy the repository URL from Azure DevOps.

#### Step 2: Azure Databricks Configuration

1. Log in to the Azure Databricks workspace.
2. Navigate to the Workspace section.
3. Create a new folder for notebooks — typically under /Workspace/Users/<username>/.
4. Configure Git integration by selecting Source Control → Git Integration.
5. Choose Azure DevOps as the Git provider.
6. Paste the ADO repository URL copied earlier.
7. Authenticate (if prompted) using your ADO credentials or a PAT.
8. A default main branch will be created in Databricks after linking.

#### Step 3: Branching and Development

1. Create a feature branch in ADO or directly from Databricks for code implementation.
2. Develop or modify notebooks within this branch in Databricks.
3. Commit changes regularly to the feature branch.
4. Verify that committed changes are automatically reflected in the ADO repository.

#### Step 4: Pull Request and Merge

1. Once development is complete, navigate to Azure DevOps → Repos → Pull Requests.
2. Create a Pull Request (PR) from the feature branch to the main branch.
3. Assign a reviewer for code approval.
4. After review and approval, merge the feature branch into the main branch.
5. The merged changes will sync back to Databricks automatically.

### 5. Validation

Verify that the latest notebooks and updates are visible in both Databricks and the Azure DevOps repository. Confirm that the pull request workflow operates as expected.

### 6. Rollback Plan

If the integration fails or causes issues:  
- Disconnect the Git integration from Databricks.  
- Reconnect using the correct repository URL or PAT.  
- Restore from the last stable version in ADO if needed.

### 7. References

* Microsoft Documentation – Connect Azure Databricks to Azure DevOps Git Repos: https://learn.microsoft.com/en-us/azure/databricks/repos/azure-devops-services
* Azure DevOps Git Repositories Overview: https://learn.microsoft.com/en-us/azure/devops/repos/git/?view=azure-devops