

# SCLM-to-Git Migration Tool

## Table of contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>What gets migrated</b>	<b>2</b>
<b>3</b>	<b>Migration process</b>	<b>2</b>
<b>4</b>	<b>Resources</b>	<b>3</b>

## 1 Introduction

Software Configuration Library Manager (SCLM) is a primarily a library manager and configuration manager. Although it also has some change management and impact analysis capabilities, the functionality is basic by today's standards. To help customers migrate from SCLM to the more modern Git software configuration management solution, IBM provides the [SCLM-to-Git Migration Tool](#).

The SCLM-to-Git Migration Tool is a set of scripts that extract SCLM data and place it in the target Git repository. The migration is performed within z/OS UNIX System Services (z/OS UNIX), using Shell scripts that subsequently invoke Groovy scripts. The SCLM-to-Git Migration Tool also creates the `.gitattributes` file for code page conversion, although the tool's generated sample build scripts are not currently compatible with the zAppBuild sample DBB build framework.

Before getting started with the migration, we recommend reviewing your current SCLM setup and removing any unused or redundant definitions to help make the migration process smoother.

## 2 What gets migrated

The SCLM-to-Git Migration Tool migrates more than just source code from SCLM to Git. The following list details what is included in the migration:

- Project definition data
  - The project definitions define the steps a source member needs to go through to be built (for example, precompile, compile, and link edit).
  - This is mainly language definitions, but also includes allocation information of any files used in the language definitions.
- Member-level metadata
  - This includes member-level overrides, which are set by an SCLM construct called an ARCHDEF. This metadata provides the ability to override compile options, change output member names, and change output file destinations.
- Generation of link decks from link-edit control (LEC) ARCHDEFs
  - LEC ARCHDEFs are how SCLM controls the compile and link-edit process. These LEC ARCHDEFs need to be converted to standard link-edit decks for use by DBB.
- Current version of the source
  - The `zimport.sh` script is used to migrate the current production version of the source to Git.
- (Optional) Previous version of the source
  - Previous versions can also be migrated by starting with the oldest version first and working backwards to the current version.

## 3 Migration process

The SCLM-to-Git Migration Tool moves source members from SCLM to Git in three phases. The steps and outputs for each phase are detailed in the tool's [Migration Process](#) documentation on GitHub. You can view a video guide for how to perform each phase in the following list.

The three-phase migration process used by the SCLM-to-Git Migration Tool:

1. Extract the SCLM metadata and source:
2. Migrate source members to a Git repository in z/OS UNIX:
3. Create sample DBB Groovy build scripts: `:::note`

The sample build scripts created in this step are not currently compatible with the zAppBuild sample DBB build framework. If you would like to use zAppBuild, you can skip this step and simply use the migrated code from the previous step (Step 2) with zAppBuild.

...

The above video guides are part of the [IBM Dependency Based Build Foundation Course](#). More information on the context, configuration, and steps for the SCLM-to-Git migration tool can be found in Module 5 (Migrating Source Members to Git) of this course.

## 4 Resources

- [SCLM-to-Git migration tool GitHub repository](#)
- [Module 5 - Migrating Source Members to Git, IBM Dependency Based Build Foundation Course](#)
- [Migration made easy - host SCLM to Git on Z \(webinar\)](#)