



MERLIN

Version: 01.00.00

HOW TO

How To Create a Merlin project from an existing GitHub repository

www.arcadsoftware.com

Document Version	:	01.00.00
Publication Date	:	June 24, 2022
Prepared by	:	FGR



North America & LATAM
70 Main Street, Suite 203
Peterborough NH 03458
USA
1-800-676-4709 (toll free)
1-603-371-9074
sales-us@arcadsoftware.com

EMEA (HQ)
55, rue Adrastée - Parc Altaïs
F-74650 Chavanod/Annecy
FRANCE
+33 450 578 396
sales-eu@arcadsoftware.com

Asia Pacific
c/o Pramex Intl Ltd
1 Austin Rd West Intl Commerce Centre
7107B 71/F Tsim Sha Tsui HONG KONG
Yau Ma tei
Hong Kong
+852 3618 6118
sales-asia@arcadsoftware.com

Table of Contents

- 1 Goal 3
- 2 List of actions to do before using Merlin 3
 - 2.1 Change type of RPGLE copies..... 3
 - 2.2 Change type of program 4
 - 2.3 Add source member for objects without source 5
 - 2.4 Type of source manage with MERLIN 6

1 Goal

This document complements the “HowTo - Create a Merlin project from an existing GitHub repository.mp4” video.

The goal of this document is to present a list of actions to do in an existing git repository before following the indications described in the video.

2 List of actions to do before using Merlin

2.1 Change type of RPGLE copies

The first thing to do is to change the **RPGLE** copies type to have the new **RPGLEINC** value.

At first, you would have a list looking like this:

Name	Last commit
..	
 APICALL.RPGLE	ARCAD Commit: Initial load of the repository
 ARTICLE.RPGLE	ARCAD Commit: Initial load of the repository
 COUNTRY.RPGLE	ARCAD Commit: Initial load of the repository
 CUSTOMER.RPGLE	ARCAD Commit: Initial load of the repository
 FAMILLY.RPGLE	ARCAD Commit: Initial load of the repository
 LOG.RPGLE	ARCAD Commit: Initial load of the repository
 PARAMETER.RPGLE	ARCAD Commit: Initial load of the repository
 PROVIDER.RPGLE	ARCAD Commit: Initial load of the repository
 VAT.RPGLE	ARCAD Commit: Initial load of the repository

MERLIN

HOW TO – v.01.00.00

At the end, after renaming the sources, you must have a list looking like this:

Name	Last commit
..	
 APICALL.RPGLEINC	Update APICALL.RPGLE
 ARTICLE.RPGLEINC	Update ARTICLE.RPGLE
 COUNTRY.RPGLEINC	Update COUNTRY.RPGLE
 CUSTOMER.RPGLEINC	Update CUSTOMER.RPGLE
 FAMILLY.RPGLEINC	Update FAMILLY.RPGLE
 LOG.RPGLEINC	Update LOG.RPGLE
 PARAMETER.RPGLEINC	Update PARAMETER.RPGLE
 PROVIDER.RPGLEINC	Update PROVIDER.RPGLE
 VAT.RPGLEINC	Update VAT.RPGLE

2.2 Change type of program

The second thing to do is to change the type of the source program in order to add the **.PGM** value in their names, because without this modification, all these sources will be considered as modules.

At first, for example for **CLLE**, you would have a list looking like this:

Name	Last commit
..	
 ORD100C.CLLE	ARCAD Commit: Initial load of the repository
 ORD100C2.CLLE	ARCAD Commit: Initial load of the repository
 ORD500C.CLLE	ARCAD Commit: Initial load of the repository
 PAR201.CLLE	ARCAD Commit: Initial load of the repository

At the end, after renaming the sources that are programs and not modules, you must have a list looking like this:

Name	Last commit
..	
 ORD100C.PGM.CLLE	Update ORD100C.CLLE
 ORD100C2.PGM.CLLE	Update ORD100C2.CLLE
 ORD500C.PGM.CLLE	Update ORD500C.CLLE
 PAR201.CLLE	ARCAD Commit: Initial load of the repository

2.3 Add source member for objects without source

The third thing to do is to create a source for each object that does not have one, in order to be able to manage them in Git.

It concerns the following types of objects: *DTAARA, *BNDDIR, *SRVPGM; *PGM (created from *MODULE and *ILESrvPGM), *DTAQ, *JOBQ, *MSGF (and sometimes *MSGD, especially if the *MSGF contains a lot of ID Message), *MSGQ, *OUTQ, *CLS and *SBSD.

It is possible to use the **AGENDOBSRC** and **AGENILESRC** ARCAD IBM i commands to generate these sources and add them later to your Git repository. You can also create them directly in the Git repository.

The name of these sources must contain the name of the object and its type.

The different types are DTAARA, BNDDIR, ILESrvPGM; ILEPGM, DTAQ, JOBQ, JOBQ, MSGF (and sometimes MSGD), MSGQ, OUTQ, CLS and SBSD.

For example:

- object **LASTORDNO** type ***DTAARA** => name of the source in Git: **LASTORDNO.DTAARA**,
- object **FARTICLE** type ***SRVPGM** => name of the source in Git: **FARTICLE.ILESrvPGM**.

The source must contain the IBM command to create the corresponding object (CRTDTAARA, CRTBNDDIR, CRTSRVPGM, ...) and you have to use the &O et &N variables for the library creation and the name of the object.

MERLIN

HOW TO – v.01.00.00

For example:

 LASTORDNO.DTAARA  57 Bytes	
1	CRTDTAARA DTAARA(&O/&N) TYPE(*DEC) LEN(6 0) VALUE(60719)

 FARTICLE.ILESRVPGM  143 Bytes	
1	CRTSRVPGM SRVPGM(&O/&N) MODULE(ART300 ART301) BNDSRVPGM(FFAMILY) +
2	ACTGRP(*CALLER) EXPORT(*SRCFILE) SRCFILE(*LIBL/QSRVSRC) SRCMBR(*SRVPGM)

2.4 Type of source manage with MERLIN

Find below the list of type of sources currently managed with MERLIN:

.BND, .BNDDIR, .PGM.CBLLE, .CBLLE, .PGM.CLLE, .CLLE, .CMD, .DSPF, .DTAARA, .ILEPGM, .ILESrvPGM, .INDEX, .LF, .MENU, .MSGF, .PF, .PNLGRP, .PRTF, .PGM.RPGLE, .RPGLE, .RPGLEINC, .PGM.SQLCBLLE, .SQLCBLLE, .SQLMASK, .SQLPERM, .SQLPRC, .PGM.SQLRPGLE, .SQLRPGLE, .SQLSEQ, .SQLTRG, .SQLUDF, .SQLUDT, .SQLVAR, .SQLXSR, .SYSTRG, .TABLE and .VIEW

If you have some sources that do not correspond to any of these types, you need to rename them.