Using the IDL Extractor for COBOL - Overview

This chapter describes how to extract IDL from a COBOL source, using the IDL Extractor for COBOL, deploy, validate and test the extraction results. IDL extraction is supported by wizards, editors and generators. This chapter covers the following topics:

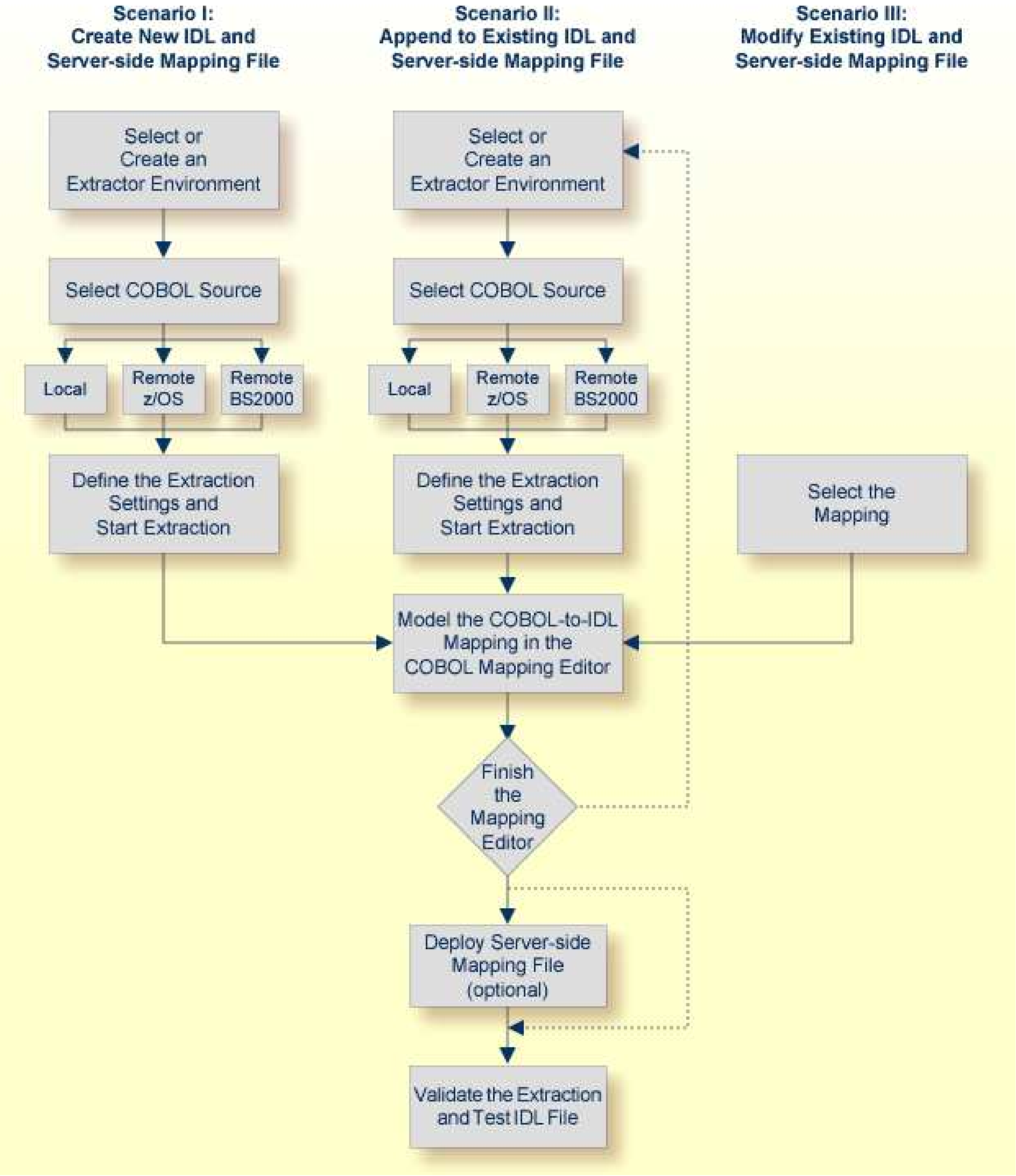
- Choosing a Scenario
- Before Starting an Extraction

Choosing a Scenario

The following scenarios are supported and are described in separate sections:

- *Scenario I: Create New IDL and Server Mapping Files*
- *Scenario II: Append to Existing IDL and Server Mapping Files*
- *Scenario III: Modify Existing IDL and Server Mapping Files*

See also *COBOL Mapping Editor*.



Before Starting an Extraction

Before you start an extraction, we recommend you first clarify the following issues:

- The interface type of your COBOL program, see *Supported COBOL Interface Types*.

- The input and output parameters of your COBOL server. Note the following:
  - COBOL REDEFINES are used in CICS as well as in batch servers. For all COBOL REDEFINES you have to clarify which redefine paths are the relevant ones for your extraction.
  - Particularly in CICS, the interface of a COBOL server is in most cases not described by the parameters given in the `PROCEDURE DIVISON` header. See *`PROCEDURE DIVISION` Mapping* and see `DFHCOMMAREA` examples under *Programming Techniques*.
- We recommend you have a basic understanding of your COBOL server, especially if you can simplify your IDL with the following:
  - Map functions of the COBOL server to IDL programs.
  - Suppress unneeded fields.
  - Map COBOL data items to constants.

The COBOL sources can contain

- copybook references; see *Copybooks* under *COBOL to IDL Mapping*
- CA Librarian (`-INC`) or CA Panvalet (`++INCLUDE`) control statements

In section *COBOL to IDL Mapping* you will find information on how the COBOL syntax is mapped to Software AG IDL using this wizard and the Mapping Editor. We recommend you read this document because it describes possibilities and alternatives for handling COBOL syntax constructs.

Make sure the COBOL source

- can be compiled with no errors and no warning
- is written in COBOL fixed format, consisting of sequence number (column 1-6), indicator area (column7), area A, (column 8-11) and area B (column 12-72) for z/OS, z/VSE, BS2000/OSD and IBM i extractions
- is either written in COBOL fixed or variable format for Micro Focus UNIX or Windows extractions and your preferences are adjusted accordingly; see *Step 2: Define the Default Settings* under *Create New Local Extractor Environment for Micro Focus (UNIX and Windows).*