



Systems Reference Library

IBM 1410 IOCS Independent Assembly Feature

This publication describes the basic programming principles of the 1410 IOCS Independent Assembly Feature, the types of assemblies made possible by this feature, and the necessary coding entries.

















CONTENTS

INTRODUCTION	USE OF THE INDEPENDENT ASSEMBLY FEATURE
Purpose of This Publication	
Talpose of This Tubilication	Types of Assemblies Possible with the Feature
Purpose of the Independent Assembly Feature 3	Coding Entries for Using the Feature
Advantages of the Feature	IOCS Assembly (Type A)
Prerequisite Publications	Object Program Assembly (Type B)
Machine Requirements	Use of DIOCS Entries for Related A and B Assemblies
	Combined Assembly of IOCS and Object Program
BASIC PRINCIPLES OF THE INDEPENDENT ASSEMBLY	(Type AB)
FEATURE 4	Additional Considerations for Assemblies using the Feature.
The Linkage Table 4	
The binning rable 4	Loading Independently Assembled Programs
Object Program Linkage to the IOCS 4	IOCS Overlay of the Priority Assignment Routines 7
File Schedulers 4	Removal of Duplicate Load Program

Purpose of this Publication

This bulletin describes the Independent Assembly Feature of the 1410 IOCS. The publication is intended to enable programmers and systems analysts to take advantage of this IOCS feature.

Purpose of the Independent Assembly Feature

The Independent Assembly Feature enables users to assemble an object program independent of the IOCS that will serve that program. Conversely, this feature permits users to assemble separately a suitable IOCS for use with any number of object programs.

Advantages of the Feature

Use of the Independent Assembly Feature saves both machine time and the programmer's time by eliminating a significant amount of reassembly work. When this feature is used, changes to an object program do not normally require reassembly of the IOCS used by that program. Also, modifications to an IOCS do not require reassembly of the object programs served by that IOCS.

In addition, use of this feature increases the flexibility of system operation. For example, routines written to handle inquiry requests from the console can be assembled with an IOCS that can also be used

by other programs. The latter can be assembled and loaded individually, each using the previously assembled IOCS and its associated inquiry routine.

Prerequisite Publications

It is assumed that the reader is familiar with those of the following publications that apply to his 1410 machine configuration:

Reference Manual, "IBM 1410 Data Processing System, " Form A24-1407

Bulletin, "IBM 1410 Autocoder," Form J24-1433 Bulletin, "IBM 1410 Input/Output Control System for Card and Tape Systems," Form J28-1432

Bulletin, "IBM 1410 Input/Output Control System for IBM 1405 Disk Storage," Form J28-0233

Bulletin, "IBM 1410 Input/Output Control System for IBM 1301 Disk Storage," Form J28-0251

Bulletin, "IBM 1410 Input/Output Control System for the IBM 1414 Input/Output Synchronizer, Models 4 and 5," Form J28-0258

Machine Requirements

The minimum machine requirements for the various routines of the IBM 1410 IOCS are specified in the publications listed above. Use of the IOCS Independent Assembly Feature does not involve additional machine requirements.

BASIC PRINCIPLES OF THE INDEPENDENT ASSEMBLY FEATURE

The Linkage Table

The linkage table consists of a series of branch instructions and subroutines that provide the linkage between independently assembled object programs and the IOCS that will serve them. This table is generated by the Autocoder processor during the independent assembly of the IOCS.

The location of the linkage table in core storage is fixed for every assembly; it always begins at location 500. The linkage table occupies up to 800 positions for an IOCS that uses the Processing Overlap and Priority special features, and up to 200 positions for an IOCS that does not use those features. The IOCS, itself, is located immediately following the linkage table, unless otherwise specified by a DIOCSORG entry.

Object Program Linkage to the IOCS

When an object program is assembled independently, IOCS macro-instructions result in branch instructions to points in the linkage table (either direct branches or branches through the object program's file schedulers). The branch instruction in the linkage table transfers control (at object time) to the appropriate IOCS routine for the original macro-instruction. For example, if the object program has a GET DISKFILE macro-instruction as a source statement, the Auto-coder processor inserts a branch instruction to that

point in the linkage table that serves as an entry to the disk GET routine of the IOCS.

The linkage table is not generated during the independent assembly of an object program but is created during the independent assembly of an IOCS. Because the points in the linkage table are constant, however, the Autocoder processor can, in effect, generate linkages from object programs to IOCS routines—even though the exact locations of the IOCS routines are not known during the assembly of the object program.

File Schedulers

Independent assembly of an object program includes the generation of file schedulers for that program, in accordance with the specifications of the DTF statements. Some IOCS macro-instructions, such as a GET for a tape file, are executed through the file schedulers and do not always require direct linkage to a separate IOCS routine. In such a case, the Autocoder processor generates a branch to the file scheduler for the execution of the macro-instruction. The file scheduler, in turn, branches to other IOCS routines (for example, an error routine) through the linkage table. Some other IOCS macro-instructions, such as a GET for a disk file, are executed by separate IOCS routines. In these instances, the Autocoder processor generates a direct branch to the linkage table, which at object time transfers control to the appropriate IOCS routine.

Types of Assemblies Possible with the Feature

Use of the Independent Assembly Feature permits the following types of assemblies:

- 1. Assembly of an IOCS with an associated linkage table. This type will be called an "A assembly."
- 2. Assembly of an object program (including associated file schedulers as specified by DTF statements). This type will be called a "B assembly."
- 3. Combined assembly of an IOCS, an associated linkage table, an object program (or routines), and file schedulers for that object program. This type will be called an "AB assembly."

NOTE: File schedulers are included in the assembly of object programs (rather than in the assembly of an IOCS) because each file scheduler is directly related to a single object program. The direct relationship of file schedulers to object programs enables each object program to define its files according to that program's particular requirements.

It should be noted that an AB assembly, because it generates a linkage table, creates an IOCS that can be used by object programs other than the one assembled with the IOCS. That is, any number of object programs produced by B assemblies can use an IOCS generated by an AB assembly that meets the object programs' requirements. It is in this respect that an AB assembly has an advantage over an assembly that does not use the IOCS Independent Assembly Feature.

The following section details the source statements necessary for each of the above types of assemblies.

Coding Entries for Using the Feature

IOCS Assembly (Type A)

The Autocoder processor generates an IOCS and its associated linkage table according to the specifications of DIOCS statements.

The programmer must first determine the IOCS requirements of all object programs that will use a particular IOCS. He must then include the necessary DIOCS statements to meet these requirements. For example, the DIOCS "CHANX" entries must specify all input/output devices for all object programs that will use the IOCS.

In addition, the programmer must include a DIOCS "OPTIONS" entry. This DIOCS entry signals the Autocoder processor that the Independent Assembly Feature is to be used. For an A assembly, the operand of this entry is LINKAGES, indicating that a linkage table is to be generated (see Figure 1).

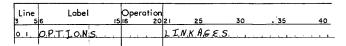


Figure 1. The DIOCS "OPTIONS LINKAGES" Entry

The last type of entry for an A assembly is a DTF Header Line with "DELETE," rather than a file name, as the operand (see Figure 2). This informs the processor that file schedulers are not to be included in the assembly, because the IOCS will be used by object programs having their own file schedulers.

NOTE: The DTF "DELETE" entry is not required if none of the object programs that will use the IOCS have DTF statements.

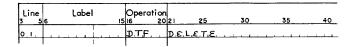


Figure 2. The DTF "DELETE" Entry

In summary, the following entries, in addition to the required Autocoder control cards, will produce an A assembly (see also Figure 3):

- 1. DIOCS Header Line.
- 2. All DIOCS entries required for the object programs that will use this IOCS.
- 3. DIOCS "OPTIONS" entry, with the operand LINKAGES.
 - 4. DTF Header Line, with the operand DELETE.

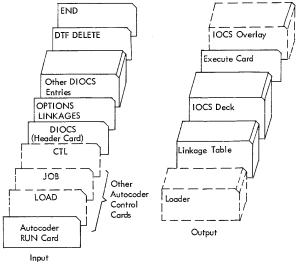


Figure 3. Card Input and Output for A-Type Assembly

Object Program Assembly (Type B)

The Autocoder processor generates an object program and its associated file schedulers according to the specifications of DIOCS and DTF statements, IOCS macro-instructions, and the source program itself. A DIOCS "OPTIONS" entry, with the operand "DELETE", must be included to signal the processor that a B assembly is to be performed (see Figure 4).

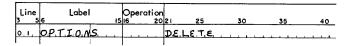


Figure 4. The DIOCS "OPTIONS DELETE" Entry

The DTF statements and IOCS macro-instructions are used as described in the IOCS publications listed in the Introduction of this bulletin. However, because the DIOCS statements are used for both A assemblies and the related B assemblies, some special considerations govern their use with the Independent Assembly Feature. The following section, "Use of DIOCS Entries for Related A and B Assemblies," contains a checklist of all DIOCS entries and explains these considerations.

It should be noted that, in general, an individual B assembly requires only those DIOCS entries that were used for the related A assembly and that apply to the object program being assembled. (For example, an object program using only tape files does not require a DIOCS for disk, even though the related A assembly included DIOCS statements for disk files to be used by other object programs.) Conversely, a B assembly cannot include any DIOCS statements that were not used for the related A assembly. Exceptions to this are noted in the DIOCS checklist.

To summarize, the following entries, in addition to the required Autocoder control cards, will produce a B assembly (also see Figure 5):

- 1. DIOCS Header Line.
- 2. All DIOCS entries required for this object program (as explained in the checklist below).
- 3. DIOCS "OPTIONS" entry, with the operand DELETE.
 - 4. DTF entries.
 - 5. Source program.

Use of DIOCS Entries for Related A and B Assemblies

The following comments apply to the use of DIOCS entries for the generation of an IOCS (A assembly) and the object programs (B assemblies) that will use that IOCS. The terminology, "same entry," as used below, indicates that the entry for the B assembly

must be a complete duplicate of the entry used for the A assembly.

DIOCS Header Line - This entry is required for all assemblies.

FEATURES - If this entry is used in A, the same entry is required for every B.

CHANx - This entry is required for A, and the same entry is required for every B.

DIOCSORG - This entry may be included in an A assembly to specify a starting point for generation of the IOCS. A DIOCSORG entry must be included in every B (to specify the starting point for the B assembly, rather than an IOCS origin). NOTE: The operand for a B assembly is determined by the results of the A assembly. For example, if the IOCS generated by the A assembly ended at location 2300, then the operand of the DIOCSORG entry for the B assembly can be no less than 2301.

PRIORITY - If this entry is used in A, the same entry is required for every B.

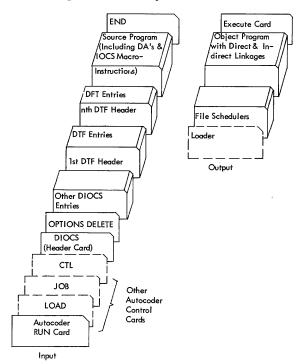


Figure 5. Card Input and Output for B-Type Assembly

CHANCHANGE - If this entry is used in A, it should be included in B only if that particular object program will perform a channel change.

READERROR - If this entry is used in A, each B can use either the same entry, or only one of the operands of the A entry, or no entry at all. However, a B entry can include only operands specified in the A entry. (If a tape unit is specified in a B entry, it must be the same unit specified in the A entry.)

ALTDRIVE CHECKPOINT COUNTS EXITS LABELDEF RWDOPTION

If any of these entries are used in A, the same entries are required for every B that uses tape files.

PROCESTYPE RNDMDEPTH STKINDEX DISKOPTION

If any of these entries are used in A, the same entries are required for every B that uses disk files.

STKAREA SGMTLENGTH NORCDEXIT DISKARMS (1405)

If this entry is used in A, the entry must be included in every B that uses disk files, but the operand for each B entry can be different from that specified in the A entry. (Zero, or any other numerical designation, used as the operand of the A entry, will prevent an "undefined label" message at assembly time. Operands of the B assemblies will be used at object time.)

DISKARMS - If this entry is used in A, the same entry is required for every B that uses disk files. NOTE: The operand is the maximum number of modules used by any object program that will be executed with the IOCS produced by the A assembly. For example, if one object program uses one module, another uses two modules, and a third object program uses three modules, then the entry should be DISKARMS 3.

INQUIRY URREQUEST

If either of these entries is used in A, the operand of the A entry must be *+1. The operand of the B entry must be the label of the user's routine (or the actual address, if the location of the user's routine is predetermined by an Autocoder Origin (ORG) statement).

NOTE: If the user's routine is assembled separately from a given object program (either in a B or an AB assembly), and if the routine and the object program will be loaded into storage together, then the B assembly of the object program must not include a DIOCS entry for the routine.

Combined Assembly of IOCS and Object Program

An AB assembly is a combination of the two types of assemblies already described. The information concerning separate A and B assemblies applies to the

AB assembly, with the following exceptions:

- 1. A separate set of DIOCS statements for the object program must not be specified. The DIOCS entries required for the object program are included in the set used to generate the IOCS and linkage table.
- 2. If DTF entries are included with the object program, the DTF "DELETE" entry must not be used.
- 3. A DIOCSORG entry cannot be used to pre-assign the object program's location. (An Autocoder Origin statement must be used for this purpose. If no Origin statement is included, the object program will be generated immediately following the IOCS.)
- 4. The DIOCS "OPTIONS DELETE" entry must not be used. (However, the DIOCS "OPTIONS LINK-AGES" entry must be included.)

ADDITIONAL CONSIDERATIONS FOR ASSEMBLIES USING THIS FEATURE

Loading Independently Assembled Programs

An IOCS and linkage table produced by either an A assembly or an AB assembly must always be loaded before the object program(s) that will use that IOCS. Also, the IOCS must be loaded each time it is to be executed with a particular object program. This is to permit re-initialization of various control fields and switches.

To load an object program immediately following the IOCS, the END Card for the A assembly must specify the address 00281 (the load program), or the Execute Card must be removed from the IOCS deck that was produced by the A assembly.

IOCS Overlay of the Priority Assignment Routine

If an A or AB assembly does not include a DIOCS "PRIORITY" entry, the Priority Assignment Routine is overlaid at object time by other IOCS routines. This overlay occurs after the first OPEN macroinstruction is encountered in the object program.

Because of this, the object program must be loaded before the IOCS routines that will overlay the Priority Assignment Routine. To enable the user to accomplish this, the A or AB assembly listing will contain a comment at the point where the object program should be inserted in the IOCS deck. This comment is, PLACE OBJECT PROGRAM DECK HERE.

Removal of Duplicate Load Program

If a load program precedes the object program deck, it must be removed when the object program is loaded immediately after the IOCS deck. This consideration also applies to an object program deck that must be inserted in the IOCS deck (as explained above).