

COBOL File Extensions Explained

Contents

Extension of INT (Intermediate Code).....	3
Extension of GNT (Generated Code).....	3
Extension of GFG (Configuration File).....	3
Executable Code.....	3
DMS Files	3
Stored Statements	4

Extension of INT (Intermediate Code)

INT is code produced when a source program is syntax-checked. Intermediate code is executable code in a Micro Focus proprietary format. An intermediate code file has the filename extension `.int`, and is often referred to as an `.int` file. INT code is used while you animate in Net Express and is not recommended for production use.

Extension of GNT (Generated Code)

GNT is a generated code file. It is an executable file containing generated code. It is produced by the compiler during the second phase of compilation, and is optimized intermediate code. Generated code files have the extension `.gnt`, and are often known as `.gnt` files. GNT code runs much faster than INT code. GNT code is production ready code.

Extension of GFG (Configuration File)

For every main executable COBOL program (such as PIPINRUN, PSPCLBLD, etc.) there will be a corresponding `.cfg` file. This file simply tells Net Express to create an `.exe` file. The `.cfg` file will be contained in the `.CBL` file directory.

Executable Code

Both `.GNT` and `.INT` code can be executed. Other types of executable code are system executable code (`.EXE`), and dynamic link libraries (`.DLL`).

DMS Files

There are often a set of DMS Scripts alongside `*.CBL` source files. These have the same file name as many/most of the COBOL programs, but with a `.dms` file extension. These scripts are run from Data Mover which load "Stored Statements" into a table called `PS_SQLSTMT_TBL`. If you edit one of these files, you will see ANSI standard SQL Data Manipulation Language (DML) commands such as Update, Select, Delete and Insert. Instead of running each `.dms` script separately, there will be some other `.dms` scripts that begin with 'STORExxx'. Each of these scripts will run all of the Stored Statements for a respective module. For example, `STOREPYI.DMS` will run all of the Stored Statements for Payroll Interface, and `STOREPT.DMS` will run the ones for PeopleTools.

Stored Statements

1. These need to be executed against EVERY database --- these load the PS_SQLSTMT_TBL table.
2. Stored statement in PS_SQLSTMT_TBL must be in synch with the source COBOL you're running. A good example of how these can get out of synch is after you've applied patches, or performed a minor upgrade.
3. Both the .CBL and .DMS programs should be maintained in the same place. Copies should always be stored in the same place and extra care taken when updating each environment. When you're running the script(s) in Data Mover, be sure to see output in the output window, and check the log file.