

For Microsoft® Windows®

# Net COBOL

## Application Distribution Guide

FUJITSU

First Edition: February 2009

The contents of this guide may be revised without prior notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Fujitsu Limited.

Copyright 2009 Fujitsu Limited.

# Preface

---

Fujitsu NetCOBOL provides you with the environment and tools to create sophisticated applications. While developing these applications, all of the supporting code and programs are available to you as part of the installed NetCOBOL product. When it is time to put your applications on other user's machines, which may not have NetCOBOL installed, it is necessary to install the appropriate runtime routines on those machines.

This manual describes the Distribution of Applications, runtimes and merge modules available to you, and explains how to use them.

## Audience

COBOL programmers responsible for building and delivering applications to users who do not have Fujitsu NetCOBOL installed on their machines.

## How This Manual is Organized

This manual contains the following information:

Chapter 1	Introduction – describes the options and issues to be dealt with when installing applications.
Chapter 2	Using the Runtime Installers – describes the runtime installers provided on the NetCOBOL download site.
Chapter 3	Using the Merge Modules – describes the Windows Installer merge modules that are available for integrating into installers based on MSI technology.

## Trademarks

InstallShield is a registered trademark or trademark of Macrovision Corporation, or other Macrovision companies, in the United States of America and/or other countries.

Microsoft, Active X, Visual Basic, Windows, Windows Vista, and Windows 2000 are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.

NetCOBOL is a trademark or registered trademark of Fujitsu Limited or its subsidiaries in the United States or other countries or in both.

Other company names and product names are trademarks or registered trademarks of individual companies.

## Conventions Used in This Manual

Example of convention	Description
<b>setup, setup</b>	Characters you enter appear in bold.
<u>Program-name</u>	Underlined text indicates a placeholder for information you supply.
ENTER	Small capital letters are used for the name of keys and key sequences such as ENTER and CTRL+R. A plus sign (+) indicates a combination of keys.
...	Ellipses indicate the item immediately preceding can be specified repeatedly.
Edit, Literal	Names of menus and options appear with the initial letter capitalized.
[def]	Indicates that the enclosed item may be omitted.
{ABC DEF}	Indicates that one of the enclosed items delimited by   is to be selected.
CHECK WITH PASCAL LINKAGE ALL PARAGRAPH-ID COBOL <u>ALL</u>	Commands, statements, clauses, and options you enter or select appear in uppercase. Program section names, and some proper names also appear in uppercase. Underlined text indicates the default.
DATA DIVISION. WORKING-STORAGE SECTION. * Get the #INCLUDE file to the data control #INCLUDE "NUMDATA.COB". PROCEDURE DIVISION.	This font is used for examples of program code.
The <i>form</i> acts as an application creation window.	Italics are occasionally used for emphasis.
"PowerCOBOL User's Guide" See Chapter 6, "Creating an Executable Program."	References to other publications or chapters within publications are in quotation marks.

# Contents

---

<b>Chapter 1. Introduction</b> .....	<b>7</b>
What are Runtime Systems?.....	7
Ways of Installing Runtime Systems.....	9
Microsoft Installer Package Files .....	9
Configuring the User Interface Level .....	10
Installing PowerCOBOL Applications .....	11
Other Installation Issues .....	12
<b>Chapter 2. Using the Runtime Installers</b> .....	<b>13</b>
<b>Chapter 3. Using the Merge Modules</b> .....	<b>15</b>
COBOL Runtime Merge Modules .....	15
PowerCOBOL Merge Module .....	15
PowerFORM Merge Module.....	15
Foundation Class Library Merge Module.....	16
Notes on the Merge Modules .....	16



# Chapter 1. Introduction

---

This chapter explains what the Fujitsu NetCOBOL runtime systems do and gives an overview of your options for installing these runtime systems.

## What are Runtime Systems?

---

Basically, a runtime system is a collection of code, usually in the form of .DLL files, required to support a particular type of program.

For example, if you use PowerFORM to create a report with various graphical features and fixed text, you write data to that report using a normal COBOL WRITE statement:

```
WRITE report-record
```

However, you are probably aware that a lot more is going to happen when that statement is executed than a simple WRITE to a file. Within PowerFORM, you create all the fixed text, specify different fonts for different fields, add graphical elements such as lines or boxes and specify where particular data should be printed on the page. When you execute a WRITE, the COBOL system needs to pass your data to special PowerFORM code that knows how to combine that data with the definition of the print form that you created using the PowerFORM utility. This special PowerFORM code is what we refer to as the PowerFORM Runtime – it is the code required to make your PowerFORM programs work at execution time.

## Fujitsu NetCOBOL Runtime Systems

NetCOBOL has five runtime systems:

- COBOL Runtime (Single-Threaded Runtime)
- PowerCOBOL Runtime
- PowerFORM Runtime
- J Adapter Class Runtime
- COBOL Server Runtime (multi-threaded runtime - no limits)

All NetCOBOL programs need either the COBOL Runtime or the COBOL Server Runtime. Additionally any application that uses J Adapter Class Generator needs the J Adapter Class runtime; any application that uses PowerCOBOL needs the PowerCOBOL runtime; and any application that uses PowerFORM needs the PowerFORM runtime.

## Why a COBOL Runtime?

You may ask, "Why do I need COBOL runtime support? Shouldn't the compiler convert all my code into just the object code necessary to execute my statements?" This is a very reasonable question. Particularly for very simple programs, it would appear to be easy for a compiler to generate a few instructions to implement the COBOL code.

The answer becomes obvious when you review in more detail the environment in which programs execute and the tasks a COBOL compiler has to perform.

Consider a program that has a single line in its PROCEDURE DIVISION:

```
MOVE A TO B
```

This program still has to be loaded into the Microsoft Windows operating system, initialized appropriately, started and stopped in the correct manner. It may also have to carry some other code to respond to messages from the Windows system – even though the programmer is making no attempt to receive or respond to those messages. For a single program, it might be reasonable to build all that code into one object module. However, when you start building 20 or 30 such programs together, it makes more sense to have all the load, initialize, start, stop etc. code built into common modules that can be called from each program. So common environment-related tasks are one reason a COBOL runtime makes sense.

Another reason is that even a simple statement like MOVE A TO B can involve a lot of work for the COBOL compiler – particularly if A and B have different data descriptions. There are hundreds of ways each of A and B can be described and thousands of possible combinations of source and target data types (for example move a PIC 99V99 COMP item to a PIC 9(10).9 item, or to a PIC 9V9(5) item or ...). Often it is better to have the compiler invoke common, highly tested routines to make the transformations, than have the compiler attempt to generate specific code for every possible move.

Thus, the best way of giving you compact, highly reliable, execution-time support for your COBOL code is to provide a runtime system that implements all the functions required by today's COBOL programs.

---

## Ways of Installing Runtime Systems

---

When you install a runtime system on a user's machine, you need to do the following:

- Place all the required .DLL (and other) files on the target machine
- Ensure the .DLL files can be found by the application program
- Make any required registry settings
- Make any required DOS PATH or environment file changes
- Register any COM objects

If you know all the appropriate settings you could in theory perform all these steps manually, or give your users instructions on how to perform these steps manually. But note, not all the information is readily available. This would be an error-prone and risky operation as some of the elements, such as the registry, can easily be corrupted by a novice user.

Consequently, Fujitsu provides runtime installers to take care of all the settings for you. You are given three choices of how to assemble your applications:

1. Create an application installer either by writing one yourself or using one of the proven installation products available (for example, InstallShield). Have this installer invoke the appropriate NetCOBOL runtime installers.
2. Use the NetCOBOL Runtime Merge Module option, described in Chapter 3, to integrate into installers based on MSI technology. Note: This option builds an installer that will work on all Windows operating systems, including Windows Vista.

---

## Microsoft Installer Package Files

---

With the release of the Windows 2000 operating system, Microsoft provided a common base for application installation to:

- Provide consistent installation rules
- Manage shared components
- Enable dynamic configuration and repair
- Provide a standard setup format

This is achieved by having all installers use a common Microsoft Installer (MSI) package file (with an .msi extension) along with common software that understands the .msi files.

Fujitsu NetCOBOL installs using the MSI technology. The NetCOBOL Wrapper uses an InstallShield interface that invokes the MSI installers, using data that is built into a single .exe along with the InstallShield interface. The runtime installation packages all use MSI technology and .msi package files.

The required MSI support is built into Windows 2000, Windows server 2003, Windows XP, Windows Vista, Windows Server 2008, and later releases. If this support is not present on the machines on which you wish to install the runtimes, you need to install it.

## Configuring the User Interface Level

When using the MSI installers you can select the level of interface seen by your users. By "user interface level" we mean the number and type of dialogs that the user sees - from seeing no dialogs, to seeing the full interface.

To select the user interface level you invoke the installer program with the appropriate command line option. The InstallShield installer programs are supplied On the NetCOBOL Download site, and are:

COBOL Runtime:	FujitsuNetCOBOL.exe
PowerCOBOL Runtime:	FujitsuPowerCOBOL.exe
PowerFORM Runtime:	FujitsuPowerFORM.exe
J Adapter Class Runtime:	FujitsuClass.exe
COBOL Server Runtime:	FujitsuNetCOBOLSVR.exe

Currently you can only configure the user interface level if you invoke these from your own installer.

The command line options are those supported by InstallShield and the Microsoft Installer. There are two relevant InstallShield options: /s and /v.

The command line format is:

```
Installer-program [/s] [/v/<msi option>]...
```

"/s" is the "silent mode" installation option which suppresses the Setup.exe initialization window. To suppress all dialogs you need to combine the /s option with the /v option.

"/v" is the option to pass options from the InstallShield program to the MSI installation program. Some relevant MSI options are documented below. Note that the "/" preceding the MSI option has to be provided, so to pass the "/qn" option you enter "/v/qn" in the command line. Also, if the MSI option contains spaces, it must be enclosed in quotes.

<msi option> Is one of the MSI installation program options. These are documented fully in Microsoft Installer documentation. Here is a brief summary of the options.

**qn** - No user interface

**qb** - Basic user interface

**qr** - Reduced user interface with a modal dialog box displayed at the end of the installation.

**qf** - Full user interface with a modal dialog box displayed at the end.

**qn+** - No user interface except for a modal dialog box displayed at the end.

**qb+** - Basic user interface with a modal dialog box displayed at the end. The modal box is not displayed if the user cancels the installation.

**qb-** - Basic user interface with no modal dialog boxes.

(Microsoft documentation contains the comment: "Please note that /qb+- is not a supported user interface level")

For example, to execute the COBOL Runtime installer with no interface you would execute the command:

```
FujitsuNetCOBOL.exe /s /v/qn
```

## Installing PowerCOBOL Applications

Simple PowerCOBOL applications can be installed by copying the application files to a machine that already has the PowerCOBOL runtime installed. More complex PowerCOBOL applications, i.e. ones that use any ActiveX controls written in PowerCOBOL or obtained from another vendor, need more attention as the ActiveX controls need to be registered to the Windows system.

For ActiveX controls created using PowerCOBOL, PowerCOBOL helps by making an installer that will register the controls for you. For ActiveX controls obtained from another vendor, or written in another language, you need to follow the instructions given by that vendor or language for registering the controls.

Your options for installing PowerCOBOL applications are outlined below in Table 1.1:

*Table 1.1. PowerCOBOL Application Installation Options*

Controls used in application	Installation method			
	Copying files	Use "Make Installer"	Use NetCOBOL Wrapper	Needs additional action
Only PowerCOBOL controls	Can install all application files	Can install all application files	Can install all application and runtime files and settings	-
ActiveX controls created using PowerCOBOL	Can only install application files.	<u>Required for registry settings. Can also install all application files</u>	Can invoke the "Make Installer" setup.exe and install all application and runtime files and settings	-
ActiveX controls created outside PowerCOBOL	Can only install applications files	Can only install application files	Can install application and runtime files and register DLL's placed in RegSrv32 folder	Register the controls.

Where:

Underlined text – indicates a method that is required, but may be used along with one or more of the other installation methods.

Using the COBOL Wrapper is the recommended route because it can be useful whichever types of control you are using. It has the added advantages of installing

the runtime support, having options to cater for other installation issues (see next section), and can also install runtime support for PowerFORM and standard COBOL applications.

## Other Installation Issues

---

You need to be aware of the following issues when installing your applications:

- **The PowerCOBOL Runtime also requires that Microsoft Data Access Component (MDAC) version 2.8.0000.00 or later is installed.**

This can be obtained from Microsoft's web site.

- **Visual C++ 2005 Runtime**

This can be obtained from Microsoft's web site.

- **You may want to uninstall Fujitsu COBOL V4/V5/V6/V7/V8 or V9 files**

If the target machine has had one or more applications installed that use Fujitsu COBOL V4/V5/V6/V7/V8/V9, you may have runtime support files installed that will conflict with the NetCOBOL V10 files. The V10 runtime provides backwards compatibility with V4/V5/V6/V7/V8/V9 so your safest option is to uninstall the V4/V5/V6/V7/V8/V9 runtimes and have a single installation of the V10 runtime. The Runtime installer has an upgrade feature, so you can upgrade Runtime to V10 if the V7 through V9 runtime is installed.

## Chapter 2. Using the Runtime Installers

---

NetCOBOL provides following runtime Installers on the NetCOBOL Download site.

COBOL Runtime:	FujitsuNetCOBOL.exe
PowerCOBOL Runtime:	FujitsuPowerCOBOL.exe
PowerFORM Runtime:	FujitsuPowerFORM.exe
J Adapter Class Runtime:	FujitsuClass.exe
COBOL Server Runtime:	FujitsuNetCOBOLSVR.exe

NetCOBOL Runtime installers are required by development package which you use to develop your application.

[NetCOBOL Standard Edition]

You need to install below runtime installer to run your application.

COBOL Runtime:	FujitsuNetCOBOL.exe
----------------	---------------------

[NetCOBOL Professional Edition]

You need to install below runtime installer to run your application.

COBOL Runtime:	FujitsuNetCOBOL.exe
PowerCOBOL Runtime:	FujitsuPowerCOBOL.exe

[NetCOBOL Enterprise Edition]

You need to install below runtime installer to run your application.

COBOL Runtime:	FujitsuNetCOBOL.exe
PowerCOBOL Runtime:	FujitsuPowerCOBOL.exe
PowerFORM Runtime:	FujitsuPowerFORM.exe
J Adapter Class Runtime:	FujitsuClass.exe

The COBOL Server Runtime is available if you bought the Enterprise Edition. This runtime installer includes the Web subroutine for the Server application. The server runtime can run multithreaded applications.

### NOTES:

- All runtimes need Microsoft Visual C++ 2005 runtime. Please get it from Microsoft Web site. And, install it before NetCOBOL Runtimes are installed.
- PowerCOBOL Runtime needs Microsoft Data Access Component (MDAC) version 2.8.0000.00 or later. Please get it from Microsoft Web site. And, install it before PowerCOBOL Runtimes are installed.



## Chapter 3. Using the Merge Modules

---

The runtime systems for NetCOBOL components are available as Windows Installer merge modules for integrating into installers based on MSI technology. This allows seamless installation of runtimes in any MSI-based installer. This section describes the different merge modules, and provides guidance for choosing the correct merge modules depending on your application. The runtimes are covered under the same EULA as other separate runtime installers, and your rights to distribute these runtimes in your installers are bound to this agreement.

**Note:** The NetCOBOL Runtime Merge Modules can be used to build an installer that will work on all Windows operating systems, including Windows Vista and later operating systems.

### COBOL Runtime Merge Modules

---

The base COBOL runtime is available in two merge modules:

**COBOLServerRuntime.msm**

**COBOLRuntime.msm**

The COBOLServerRuntime.msm is the multi-threaded COBOL runtime.

**Note:** This runtime may be distributed only if you are licensed for NetCOBOL Enterprise Edition.

The COBOLRuntime.msm is the basic COBOL runtime (non-multithreaded).

These runtime modules are not designed to be distributed together in a single installer. Install only one runtime module, based on the version that you are licensed to distribute.

### PowerCOBOL Merge Module

---

Installers for PowerCOBOL applications should include the following merge module:

**PowerCOBOLRuntime.msm**

**NOTES:**

- The COBOL runtime must be installed in addition to the PowerCOBOL runtime.
- The PowerCOBOL Runtime needs Microsoft Data Access Component (MDAC) version 2.8.0000.00 or later. This can be obtained from Microsoft's web site. It must be installed before PowerCOBOL Runtimes are installed.

### PowerFORM Merge Module

---

Installers for PowerFORM applications should include the following merge module:

**PowerFORMRuntime.msm**

**NOTE:** The COBOL runtime must be installed in addition to the PowerFORM runtime.

## J Adapter Class Library Merge Module

---

Installers for applications that use the J Adapter Class Library should include the following merge module:

### **JAdapterClassLibrary.msm**

**NOTE:** The COBOL runtime must be installed in addition to the J Adapter Class Library components.

## Notes on the Merge Modules

---

- Each runtime file is installed in the common files folder shown below if the [TARGET] folder is not defined.  
[Common Files]\Fujitsu\COBOL
- The runtime merge modules use the same component codes as the stand-alone NetCOBOL runtime installers, so each will be able to detect and install correctly if the other already exists on the target system. Therefore, your installer does not need to detect whether the NetCOBOL for Windows V9 runtime is installed, and can simply always include the merge module.

Note that prior versions of NetCOBOL for Windows are not installed in this manner, so you will need to uninstall any earlier versions of the runtime before installing the V9 version of the runtime.

The stand-alone runtime installers are listed below, and can coexist with the NetCOBOL merge modules.

COBOL Runtime:	FujitsuNetCOBOL.exe
PowerCOBOL Runtime:	FujitsuPowerCOBOL.exe
PowerFORM Runtime:	FujitsuPowerFORM.exe
J Adapter Class Runtime:	FujitsuClass.exe
COBOL Server Runtime:	FujitsuNetCOBOLSVR.exe

- All runtimes need Microsoft Visual C++ 2005 runtime. This can be obtained from Microsoft's web site. It must be installed before NetCOBOL Runtimes are installed.

For more information on using merge modules based on Windows Installer technology, refer to the Windows Installer documentation available from Microsoft.