

MicroFocus UNIX Runtime Component

Release 8.1.3

November 2013



IKAN Solutions N.V.
Kardinaal Mercierplein 2
B-2800 Mechelen
BELGIUM

Copyright © 2013, IKAN Solutions N.V.

No part of this document may be reproduced or transmitted in any form or by any means, electronically or mechanically, for any purpose, without the express written permission of IKAN Solutions N.V.

MetaSuite, MetaStore Manager, MetaMap Manager and Generator Manager are trademarks of IKAN Solutions N.V.

DB2/PE and DB2/6000 are trademarks of International Business Machines.

Oracle is a trademark of Oracle Corporation.

Sybase is a trademark of Sybase, Inc.

Informix is a trademark of Informix, Inc.

Table of Contents

Chapter 1 - Introduction	1
1.1. Related Products	1
1.2. Terminology.....	1
1.3. System Requirements	1
1.4. Pre-installation Requirements.....	2
Chapter 2 - About This Manual.....	3
2.1. Prerequisites	3
2.2. Related Publications	3
Chapter 3 - MetaSuite Generator Runtime Installation.....	5
3.1. Runtime Components.....	5
3.2. UNIX Environment.....	5
3.3. Install Runtime	5
3.4. Compile & Link Runtime.....	6
Chapter 4 - Compile a MetaSuite Generator Program	7
4.1. Compilation	7
4.2. Pre-compilation	8
<i>DB2/6000 and DB2/PE</i>	8
<i>Informix</i>	8
<i>Oracle 7</i>	8
<i>Oracle 8</i>	9
<i>Sybase</i>	9
Chapter 5 - Run a MetaSuite Generator Program	11
5.1. Example	11
Chapter 6 - RDBMS Run Considerations.....	12
6.1. DB2/6000.....	12
6.2. Informix.....	12
6.3. Oracle	12
6.4. Sybase.....	13
6.5. ODBC.....	13

Introduction

1.1. Related Products

- MetaStore Manager (and the corresponding batch component MSBSTORE)
- MetaMap Manager (and the corresponding batch component MSBMAP)
- Generator Manager (and the corresponding batch component MSBGEN)

1.2. Terminology

MDL	MetaSuite Definition Language
MXL	MetaSuite Export Language
MGL	MetaSuite Generated Language
MRL	MetaSuite Run Language

CBL	COBOL source
COB	COBOL source
SQB	SQL COBOL source
ECO	Embedded SQL COBOL source
<Ins>	MetaSuite installation folder on the client side

1.3. System Requirements

CPU	Pentium Processor or higher
System RAM	Minimum of 96 MB
Hard disk space	Minimum 300 MB of free disk space for storage of MetaSuite software and .NET Framework.

Operating System	<ul style="list-style-type: none">• Windows Vista• Windows Seven• Windows XP with Service Pack 3
Software	<ul style="list-style-type: none">• Microsoft .NET Framework 2.0 (already included in Windows Vista)• File transfer program (e.g. PC Support)• MT9750 terminal emulator

1.4. Pre-installation Requirements

Before installing the runtime component, you must:

1. Install MetaSuite and select *MicroFocus on UNIX* as Generator environment.

For more detailed information, refer to the *Installation and Setup Guide*.

2. Create the MetaSuite Generator Dictionary for MicroFocus on UNIX.

For more detailed information, refer to the *Generator Manager User Guide*.

About This Manual

This guide describes how to install the MetaSuite MicroFocus UNIX runtime component. More specifically, it describes the installation of the following MetaSuite components:

- Base product
- MetaSuite Database Interfaces

The instructions for these components refer to additional information found in separate documents. Be sure to have those documents available during the installation.

2.1. Prerequisites

Product installers are expected to be familiar with their host operating systems and software installation processes.

2.2. Related Publications

The MetaSuite User and Reference Guides describe the different MetaSuite components and provide examples for using MetaSuite. Those guides should be available for reference during the installation and test procedures described here.

The following table gives an overview of the complete MetaSuite documentation set.

Release Information	Release Notes 8.1.3
Installation Guides	<ul style="list-style-type: none"> • BS2000/OSD Runtime Component • DOS/VSE Runtime Component • Fujitsu Windows Runtime Component • MicroFocus Windows Runtime Component • MicroFocus UNIX Runtime Component • OS/390 and Z/OS Runtime Component • OS/400 Runtime Component • VisualAge Windows Runtime Component • VisualAge UNIX Runtime Component • VMS Runtime Component
User Guides	<ul style="list-style-type: none"> • INI Manager User Guide • Installation and Setup Guide • Introduction Guide • MetaStore Manager User Guide • MetaMap Manager User Guide • Generator Manager User Guide

Technical Guides	<ul style="list-style-type: none"> • ADABAS File Access Guide • IDMS File Access Guide • IMS DLI File Access Guide • RDBMS File Access Guide • XML File Access Guide • Runtime Modules • User-defined Functions User Guide
------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

If you are unfamiliar with MetaSuite, the following technical description provides you with a brief overview.

The MetaSuite System	MetaSuite is designed for data retrieval, extraction, conversion and reporting. It includes a workstation-based graphical user interface and a mainframe runtime component.
MetaSuite Database Interfaces	MetaSuite can access data from a number of database management systems, using the same commands, program structure and retrieval techniques used for non-database files. Each database interface is available as an optional enhancement to the base product.
MetaMap Manager	MetaMap Manager is the MetaSuite tool used to define models. Such models are intuitively built by describing overall program specifications, input file definitions (data and process) and target file definitions (data and process).
MetaStore Manager	MetaStore Manager is a tool that provides metadata maintenance and documentation services.
Generator Manager	The Generator Manager is the system administration tool. All kinds of basic functionalities and customization possibilities are supported by this tool.

MetaSuite Generator Runtime Installation

3.1. Runtime Components

The `x:\installdirectory\GENMicroFocus_UNIX\system` directory contains the following files:

```
COBOL sources:      MS*813.cbl
UNIX scripts:      mkcob
                   mkcobdb2
                   mkcobinf
                   mkcбора7
                   mkcбора-8-9
                   mkcобsyb
                   mkrts813
                   msrun
```

3.2. UNIX Environment

Create a MetaSuite UNIX account called "metasuit" with the following directories:

```
$HOME/user
$HOME/system
```

3.3. Install Runtime

Use a File Transfer Program (FTP) to copy the MetaSuite runtime source code (all the .cbl files) from the `MetaSuite\GenMicroFocus_Unix\SYSTEM` folder into the MetaSuite source library.

Note: If your COBOL compiler does not support Unicode (i.e., the Unicode functions "display-of" and "national-of" with the option "second argument" to provide a code page (for example 1208 to do the conversion to or from UTF-8)), some runtime programs can not be compiled.

Note: Make sure that files are transferred case sensitive and as text files (ASCII CRLF).

3.4. Compile & Link Runtime

Following needs to be executed in the UNIX environment to compile and link the runtime:

```
cd $HOME/system
chmod +x uxppt10
chmod +x mk*
uxppt10
```

The following procedure will be executed:

Note: ??? replaces the 3 middle characters of the COBOL source files.

```
# make MetaSuite runtime library
rm libms813.a
rm ms???813.o
cob -xc ms???813.cbl
rm ms???813.int
ar -rs libms813.a ms???813.o
rm ms???813.o
```

Note: The scripts used in the following sections are for documentary purposes only.

Compile a MetaSuite Generator Program

Following compilation scripts are transferred to your UNIX environment, which will be used for (pre-) compilation of your generated MetaSuite programs.

```
mkcob
mkcobdb2
mkcobinf
mkcbora7
mkcbora8
mkcobsyb
```

Note: The scripts used in the following sections are for documentary purposes only.

4.1. Compilation

Transfer the `x:\installdirectory\GENMicroFocus_UNIX\MGL\MXL-name.mgl` to `$HOME/user/MXL-name.cbl`

```
cd $HOME/user
mkcob MXL-name
```

The following procedure will be executed:

```
echo "Compiling:"
echo $1
rm $1
cob -k $1.mgl -x -o $1 -P -L $HOME/system -lms813
rm $1.o
rm $1.int
```

4.2. Pre-compilation

When your MetaSuite program accesses a RDBMS SourceFile, the generated program must be pre-compiled before compilation. Pre-compilation is dependent of the type of RDBMS that will be accessed.

DB2/6000 and DB2/PE

Transfer the `x:\installdirectory\GENMicroFocus_UNIX\MGL\MXL-name.mgl` to `$HOME/user/MXL-name.sqb`

```
cd $HOME/user
mkcobdb2 MXL-name
```

The following procedure will be executed:

```
COBDIR=/usr/lib/COBOL
PATH=/usr/lib/COBOL/bin:$PATH
# Connect to a database.
db2 connect to *database-name* user *userid* using
*password*
# Precompile the program.
db2 prep $1.sqb bindfile target mfcob datetime iso messages $1.msg
# Compile the program.
cob -x$1.cbl -L/home/db2se/sqlllib/lib -ldb2 -ldb2gmf -P -
L $HOME
system -lms813
# Bind the program to the database.
db2 bind $1.bnd blocking all datetime iso isolation cs
rm $1.o $1.int $1.cbl
# Disconnect from the database.
db2 connect reset
```

Informix

Transfer the `x:\installdirectory\ GENMicroFocus_UNIX \MGL\MXL-name.mgl` to `$HOME/user/MXL-name.eco`

```
cd $HOME/user
mkcobinf MXL-name
```

The following procedure will be executed:

```
export COBDIR=/opt/lib/COBOL
export INFORMIXDIR=/usr/informix
export PATH=$COBDIR/bin:$INFORMIXDIR/bin:$PATH
rm $1
esqlCOBOL -t mf2 -log $1.log -esqlout $1.cob -native
$1.eco
rm $1.o $1.int
```

Oracle 7

Transfer the `x:\installdirectory\ GENMicroFocus_UNIX \MGL\MXL-name.mgl` to `$HOME/user/MXL-name.mgl`

```
cd $HOME/user
mkcbora7 MXL-name
```

The following procedure will be executed:

```
COBDIR=/opt/lib/COBOL
PATH=/opt/lib/COBOL/bin:$PATH
rm $1
procob userid=*userid*/password* iname=$1.cbl
oname=$1.cob mode=ansi litedelim=apost sqlcheck=full
cob -x $1.cob -P $ORACLE_HOME/lib/osntab.o -L
$ORACLE_HOME/lib/lcobsq1 -lsq1 -lsq1net -lora -lsq1net -
lora -lnlsrtl -lc3v6 -lcore3 lnlsrtl -lcore3 -lc -lm -
lmproc -L $HOME/system -lms813
rm $1.o
rm $1.int
```

Oracle 8

Transfer the x:\installdirectory\ GENMicroFocus_UNIX \MGL\MXL-name.mgl to \$HOME/user/MXL-name.mgl

```
cd $HOME/user
mkcbora8 MXL-name
```

The following procedure will be executed:

```
rm $1
procob userid=*userid*/password* iname=$1.mgl oname=$1.cob mode=ansi
sqlcheck=full
cob -x -o $1 $1.cob -L$ORACLE_HOME/lib/ \
$ORACLE_HOME/precomp/lib/cobsq1intf.o \
-lclntsh -lclient \
$ORACLE_HOME/lib/libsql.a \
$ORACLE_HOME/lib/scorept.o \
$ORACLE_HOME/lib/sscoreed.o \
$ORACLE_HOME/rdbms/lib/kpudfo.o \
$ORACLE_HOME/lib/nautab.o \
$ORACLE_HOME/lib/naeet.o \
$ORACLE_HOME/lib/naect.o \
$ORACLE_HOME/lib/naedhs.o \
-lnetv2 -lnttcp -lnetwork -lnr -lnetv2 -lnttcp -lnetwork \
-lclient -lcommon -lgeneric -lmm -lnlsrtl3 -lcore4 \
-lnlsrtl3 -lcore4 -lnlsrtl3 -lnetv2 -lnttcp -lnetwork \
-lnr -lnetv2 -lnttcp -lnetwork -lclient -lcommon \
-lgeneric -lepc -lnlsrtl3 -lcore4 -lnlsrtl3 -lcore4 \
-lnlsrtl3 -lclient -lcommon -lgeneric -lnlsrtl3 -lcore4
-lnlsrtl3 -lcore4 -lnlsrtl3 -lnsl -lsocket -lgen -lthread \
-lm -L $HOME/system -lms813
rm $1.o
rm $1.int
```

Sybase

Transfer the x:\installdirectory\ GENMicroFocus_UNIX\MGL\MXL-name.mgl to \$HOME/user/MXL-name.mgl

```
cd $HOME/user
mkcobsyb MXL-name
```

The following procedure will be executed:

```
rm $1
cobpre -Cmf_word -Sserver -User -Ppassword -Ddatabase -b -
KSEMANTIC -L$1.lis -O$1.cob $1.mgl
echo Compiling $1 ...
cob -x -P -C IBMCOMP -o $1 $1.cob -P \
    -L $SYBASE/lib -lcobct -lct -lcs -lcomn -lintl -ltcl -lm \
    -L $HOME/system -lms813
rm $1.o $1.int
echo mkcobsyb $1 ended
```

Run a MetaSuite Generator Program

When a MetaSuite Generator program is run, there is a template run script generated as well, which is stored in `x:\installdirectory\GENMicroFocus_UNIX\MRL\MXL-name.MRL`. You can customize the generation of the template run script by changing the code tables, which are used for the template run script. Refer to the *Generator Manager User Guide* for more information on how the code tables for the template run scripts are built.

Transfer the `x:\installdirectory\GENMicroFocus_UNIX\MRL\MXL-name.mrl` to `$HOME/user/MXL-name.mrl`.

Note: The scripts used in the following sections are for documentary purposes only.

5.1. Example

The following is an example run script, which is generated for EX0.MXL.

```
# Unix Runscript For ex0-0001
rm ex0.log
rm ex0.lst
rm ex0.d01
export dd_PPTIPT=NUL
export dd_PPTLID=NUL
export dd_PPTF01=employee-master
export dd_PPTLOG=ex0.log
export dd_PPTLST=ex0.lst
export dd_PPTTD01=ex0.d01
export COBSW==+D
ex0
echo $?
```

You will be able to run this script after the following steps:

```
cd $HOME/user
chmod +x EX0.mrl
```

RDBMS Run Considerations

Users can set only 1 runtime variable on the command for security reasons. The next example will set the password:

```
XXXXXXXXX.exe SYS-DB-PASSWORD = 'password'
```

6.1. DB2/6000

DB2/6000 users should supply the following MetaSuite Generator application runtime variable in the PPTIPT file to connect to a specific database:

```
SYS-DB-CONNECT = 'server'
```

6.2. Informix

DB2/6000 users should supply the following MetaSuite Generator application runtime variable in the PPTIPT file to connect to a specific database:

```
SYS-DB-CONNECT = 'server'
```

6.3. Oracle

Oracle users should supply the following MetaSuite Generator application runtime variable in the PPTIPT file to connect to a specific database:

```
SYS-DB-CONNECT = 'connect string'  
SYS-DB-USER = 'user-id'  
SYS-DB-PASSWORD = 'password'
```

You can use as well the transparent logon for Oracle:

```
SYS-DB-CONNECT = '(AUTO)'
```

6.4. Sybase

Oracle users should supply the following MetaSuite Generator application runtime variable in the PPTIPT file to connect to a specific database:

```
SYS-DB-CONNECT = 'connect string'
SYS-DB-USER = 'user-id'
SYS-DB-PASSWORD = 'password'
```

You can use as well the transparent logon for Oracle:

```
SYS-DB-CONNECT = '(AUTO)'
```

6.5. ODBC

If the SQL dialect is set to ODBC then users should supply the following MetaSuite Generator application runtime variable in the PPTIPT file to connect to a specific database:

```
SYS-DB-CONNECT = {'ODBC database name' | '(DBNAME)'}
SYS-DB-USER = 'user-id'
[SYS-DB-PASSWORD = 'password']
```

Special rule:

If SYS-DB-CONNECT = '(DBNAME)' then the DBNAME parameter value (specified in the ADD FILE command in the MDL) will be taken. The runtime variable SYS-DB-DATABASE however can overwrite this value.

So the logic order is:

```
SYS-DB-CONNECT
SYS-DB-DATABASE (if SYS-DB-CONNECT='(DBNAME)')
File specific DBNAME (if SYS-DB-DATABASE empty)
```