
Fast/Unload Release Notes Version 4.1

September, 2003



Sirius Software, Inc.
875 Massachusetts Avenue, Suite 21
Cambridge, MA 02139

Telephone: (617) 876-6677

FAX: (617) 234-1200

E-mail: support@sirius-software.com

World Wide Web: <http://sirius-software.com>

November 17, 2004

© 2004 Sirius Software, Inc.

Proprietary Notices

The following products:

- *Fast/Unload*
- *Fast/Unload User Language Interface*
- *Sirius Mods*

are proprietary products of Sirius Software, Inc.:

Sirius Software, Inc.
875 Massachusetts Avenue, Suite 21
Cambridge, Massachusetts 02139
USA

Model 204™ is a proprietary product of Computer Corporation of America:

Computer Corporation of America
500 Old Connecticut Path
Framingham, Massachusetts 01701
USA

Contents

Proprietary Notices	ii
Contents	iii
Chapter 1: Introduction	1
Summary of significant changes	1
Chapter 2: Maintenance and Support	3
Changed or new #functions	3
Change to display of program listing	3
New Documentation tool	3
Chapter 3: New Features	5
Performance enhancements	5
New or changed program parameters	5
New and modified FUEL statements	6
#IF, #ELSEIF, #ELSE, and #END IF "preprocessor" statements	6
Multiple output stream support	7
New or changed #functions	7
Miscellaneous enhancements	8
Chapter 4: Compatibility/Fixes	9
Backwards compatibility with Fast/Unload 4.0	9
Fixes in Fast/Unload 4.1 but not in 4.0	9
Version co-requisites	9

CHAPTER 1 ***Introduction***

This document lists the enhancements and other changes contained in the newest release of *Fast/Unload*, version 4.1. The previous generally available version of *Fast/Unload*, 4.0, was released in January, 2000.

1.1 Summary of significant changes

Fast/Unload version 4.1 introduces the following significant changes:

- Support for multiple output streams. See [“Multiple output stream support”](#) on page 7.
- Support for very large FUEL programs. See [“Performance enhancements”](#) on page 5.
- #IF, #ELSEIF, #ELSE, and #END IF "preprocessor" statements. See [“New and modified FUEL statements”](#) on page 6.
- Enhancements for easier debugging of FUEL programs. See [“Maintenance and Support”](#) on page 3.

CHAPTER 2 *Maintenance and Support*

These enhancements to *Fast/Unload* do not affect its intrinsic functionality, but they do affect the way Sirius Software delivers maintenance and support for the product.

2.1 Changed or new #functions

#ABDUMP ABEND the *Fast/Unload* job, with a specified condition code. This can be used under the direction of the Sirius Software support staff.

2.2 Change to display of program listing

The statement offset of the generated code is displayed on each line of the listing of the FUEL program. This may be useful to Sirius Software support.

2.3 New Documentation tool

All the text in the Sirius reference documentation is included in the Sirius Master Index, a searchable database you can download from the Documentation page of the Sirius web site,

The download package contains a zipped collection of the Sirius PDF documents along with the Adobe-built, English-only, index (a PDX file and its supporting files), which you view with the Adobe Acrobat Reader. The index tool offers a variety of searching options and displays results in a linked set per document.

CHAPTER 3 *New Features*

3.1 Performance enhancements

The following changes help you to achieve improved performance with *Fast/Unload*:

Large FUEL programs In *Fast/Unload* version 4.0, the size of a FUEL program was limited, based on a maximum size of the generated code of about 4 million bytes. This limit has been removed. (This feature was also implemented as part of the maintenance zaps to *Fast/Unload* version 4.0, but it was not previously mentioned in any *Release Notes*.)

The limit to the size of a FUEL statement has also changed: the default has been more than tripled, and you can increase the limit with the new LIBUFF program parameter. (An increase to the limit was also implemented as part of the maintenance zaps to *Fast/Unload* version 4.0, but it was not previously mentioned in any *Release Notes*.)

3.2 New or changed program parameters

The following new parameters may be passed to *Fast/Unload*:

FNVMASK X"yz" FNVMASK specifies which of the 8 characters of the name of the file being unloaded may differ from the file name stored on the disk pages of the *Model 204* file.

If *Fast/Unload* is invoked via the *Fast/Unload User Language Interface*, the value of the FNVMASK parameter in *Model 204* is automatically communicated, and this is the default value of FNVMASK for the unload. Invoking the *Fast/Unload User Language Interface* with a non-zero value of FNVMASK in *Model 204* requires the use of version 6.1 or later of the *Sirius Mods*.

This parameter was also implemented as part of the maintenance zaps to *Fast/Unload* version 4.0, but it was not previously mentioned in any *Release Notes*.)

LIBUFF n LIBUFF may be used to increase the limit on the length of a FUEL statement.

SORTOUTD

SORTOUTD has been added as a third option (along with FUNOUT and SORTOUT) to control output from sort(s) invoked by *Fast/Unload*. These three parameters are mutually exclusive, and SORTOUTD is the default.

- If FUNOUT is set on:
Fast/Unload will handle all output from invoked sorts, sending each stream to the destination (DDNAME, FILEDEF) specified in the declaration of the stream (OUT TO *dest* or UAI TO *dest*). For legacy FUEL programs (that is, one output stream, not explicitly declared), output will go to FUNOUT.
- If SORTOUT is on:
For legacy UAI programs, the sort itself will send output to FUNOUT. For legacy non-UAI programs, the sort will send output to SORTOUT.

For non-legacy programs, the sort will do the output for each sorted output stream; the name of each particular output stream will be sent to the particular sort by an internally generated OPTION SORTOUT= *dest* sort control statement.

- If SORTOUTD is on:
This option works like SORTOUT except for legacy non-UAI programs, for which the sort will send output to FUNOUT, not SORTOUT, consistent with the handling for UAI sorted output.

3.3 New and modified FUEL statements

This section describes new statements for use in *Fast/Unload* as well as existing statements that have new options.

3.3.1 #IF, #ELSEIF, #ELSE, and #END IF "preprocessor" statements

#IF through #END IF statement blocks allow the FUEL programmer to write programs that work across a variety of different files, even if all the files do not contain exactly the same fields.

#IF <fieldname> { DEFINED | UNDEFINED } tests the currently open file; subsequent FUEL statements are compiled if the #IF statement is true, that is, if the specified <fieldname> is defined (or not defined) in the currently open file.

#ELSEIF <fieldname> { DEFINED | UNDEFINED } statements optionally follow the #IF. An optional #ELSE statement is always true.

A #IF block must end with a #END IF statement. #IF blocks may not be nested.

The first #IF / #ELSEIF / #ELSE block that is true causes all FUEL statements to be compiled up to the next #ELSEIF / #ELSE / #END IF block. All other statements following such a true #ELSEIF / #ELSE / #END IF block are skipped until the #END IF is reached.

3.3.2 Multiple output stream support

```
OUT TO destination [DEF[AULT]]
UAI [TO destination [DEF[AULT]]] ...
SORT [TO destination] ...
[TO destination] PUT ...
[TO destination] OUTPUT
[TO destination] PAI
[TO destination] UNLOAD[C] [...]
... #RECOUT[(destination)] ...
... #OUTLEN[(destination)] ...
... #OUTPOS[(destination)] ...
```

In the statements shown above, the OUT TO *destination* and UAI TO *destination* serve as declarations specifying the name of an output stream. This name is used in the TO *destination* clause on subsequent statements that actually generate the output.

The stream name is also used as the parenthesized qualifier on the special variables #RECOUT, #OUTLEN, and #OUTPOS. And similarly, the SORT [TO *destination*] qualifier associates the sort control statement to a particular stream.

In addition, a SORTOUTD parameter is now available as a third option (along with FUNOUT and SORTOUT) to control output from sort(s) invoked by *Fast/Unload*. SORTOUTD is described further in [“New or changed program parameters” on page 5](#)

See the ***Fast/Unload Reference Manual*** for the details of *Fast/Unload* multiple output stream support.

3.4 New or changed #functions

See [“Changed or new #functions” on page 3](#) for #functions that are used for maintenance purposes.

3.5 Miscellaneous enhancements

Nesting level Whenever the nesting level changes within a FUEL program, the new level is displayed on the program listing, along with an indication whether it is increasing or decreasing (or, in some cases, creating a block of code with neither an increase nor decrease).

This section lists any compatibility issues with *Fast/Unload*, and any fixes contained in this version of *Fast/Unload* but not, as of the date of this release, in the immediately prior version (4.0).

4.1 Backwards compatibility with Fast/Unload 4.0

This section lists any differences in processing that result from execution with *Fast/Unload* version 4.1, as compared with the same inputs to *Fast/Unload* version 4.0 at current maintenance levels.

SORTOUT, SORTOUTD parameters

In version 4.0, SORTOUT was the default. In version 4.1, SORTOUTD is the default.

4.2 Fixes in Fast/Unload 4.1 but not in 4.0

This section lists other fixes to features existing in *Fast/Unload* version 4.0 but which, in absence of customer problems, have not, as of the date of the release, been fixed in that version.

There are no problems without fixes.

4.3 Version co-requisites

This section lists any restrictions on usage of various products (including *Fast/Unload* itself) which will be imposed by use of version 4.1 of *Fast/Unload*.

There are no version co-requisites.

