Genera 8.0 Release Notes

Genera 8.0: Introduction and Highlights

Genera 8.0 is a unified release for Ivory-based and 3600-family machines. It includes an implementation of the new Common Lisp Object System (CLOS), as well as numerous other customer-suggested feature enhancements, performance improvements, and bug fixes. It also incorporates improvements from Genera 7.3 Ivory and Genera 7.4 Ivory, previously unavailable to users of 3600-family machines.

Anyone currently using Genera 7.2, Genera 7.4 Ivory, or Genera 7.4 Ivory ECO #3 (UX400S only) can upgrade easily to Genera 8.0. Source code from those releases is fully upward compatible. Programs running on Ivory-based machines will run as is in Genera 8.0, although we recommend recompiling them to take advantage of performance improvements and a bug fix (see the section "Miscellaneous Improvements to the Lisp Language in Genera 8.0").

Programs running on 3600-family machines will run in Genera 8.0 after recompilation. For more information, see the document *Upgrading to Genera 8.0*.

The *Genera 8.0 Release Notes* describe changes made since Genera 7.4 Ivory. In addition, many chapters in this document contain a section, "Changes to ... Since Genera 7.2", for users of 3600-family machines migrating from Genera 7.2. These sections list changes made to Genera in Genera 7.3 Ivory and Genera 7.4 Ivory.

If you have MACSYMA 416.15, you can run it with Genera 8.0. You need not postpone upgrading to 8.0 to wait for a new MACSYMA release. See the section "Running MACSYMA in Genera 8.0" for notes about loading MACSYMA if you have a 3600-family machine.

A full list of changes in Genera 8.0 appears in the Table of Contents. These release notes cover the following topics:

"Changes to the Lisp Language in Genera 8.0"

> The most significant change to Lisp in this release is the addition of Symbolics CLOS, our implementation of the Common Lisp Object System. For more information, see the section "CLOS is Available in Genera 8.0".
>
> Genera 8.0 also includes the Common Lisp Developer, a tool that aids in developing and testing Common Lisp programs on a Symbolics system.

"Changes to Utilities in Genera 8.0"

> Changes include:
>
> - A new mode of garbage collection, In-Place garbage collection.

- Improvements to the Conversion Tools that were first released with Genera 7.1. These tools are now more powerful and less tedious to use, and can do Flavors to CLOS conversion.

"Changes to the User Interface in Genera 8.0"

A number of new Command Processor commands have been added in Genera 8.0. Many of them are useful for remote terminal use.

"Changes to Zwei and Zmacs in Genera 8.0"

New commands and key bindings are now available.

"Changes to Zmail in Genera 8.0"

A number of Zmail bugs have been fixed in this release.

"Changes to Files and Streams in Genera 8.0"

The most noticeable change is that character I/O streams are faster and more reliable in Genera 8.0. In addition, there is a new facility for compressing and uncompressing distribution tapes.

"Changes to User Interface Programming Facilities in Genera 8.0"

Genera 8.0 provides the foundation for CLIM, the new Common Lisp User Interface Management System standard, which will be available from Symbolics as an optional layered product a few months after this release. This foundation is invisible to users of Genera 8.0.

"Changes to Site Operations in Genera 8.0"

Significant changes include:

- Genera 8.0 allows machines with IP-TCP loaded in their worlds to be booted using the Internet network as their primary network.

- The software installation procedure for the Symbolics UX400S has been automated and greatly simplified. For more information, see the section "Tape Installation Instructions for Symbolics UX Machines".

- The Symbolics UX400S is now customer-installable. For more information, see the section "Symbolics UX-Family Hardware Installation Guide".

"Changes to the FEP in Genera 8.0"

The recommended division of contents between hello.boot and boot.boot files has changed.

"Changes to Networks in Genera 8.0"

Changes include:

- Many networking and I/O functions have been reimplemented to provide more robust and faster performance.

- The Network File System (NFS), X Window System client, and support of Postscript hardcopy are much faster and more reliable.

- TCP/IP performance has been improved.

- Dialnet registries are now obsolete.

- Dialnet has been made more robust.

- The Remote Procedure Call (RPC) protocol is now available on all Symbolics systems. (It is separately loadable on 3600-family machines.)

"Changes to Genera 8.0 for MacIvory"

Changes include:

- Zmail can now read and write mail files in the Macintosh file system

- The Genera screen behaves more like a Macintosh window.

"Changes to Genera 8.0 for the Symbolics UX-Family"

A number of bugs have been fixed, and new features have been added.

"Changes to the Genera Documentation Set in Genera 8.0"

Changes include:

- A new tutorial in Genera programming techniques, *Program Development Tutorial*. It replaces the "Program Development Tools and Techniques" tutorial that was located in Book 4 of the Genera 7 documentation set.

- A comprehensive new *System Index* that includes entries for all Genera and Ivory system documents.

- Sections relocated to different printed documents to facilitate the lookup of related topics.

"Notes and Clarifications in Genera 8.0"

This section contains explanations and clarifications of items that people found confusing in previous releases and documentation.

Changes to the Lisp Language in Genera 8.0

Changes to the Lisp Language Since Genera 7.2

Square Root Functions Handle Denormalized Fractions Correctly

In Genera 7.2, denormalized arguments to **zl:sqrt** and **sqrt** produced answers that were not as accurate as they should have been.

(Denormalized numbers are those floating-point numbers with magnitudes less than least-positive-normalized-single-float => 1.1754944e-38 or least-positive-normalized-double-float => 2.2250738585072014d-308.)

For example, **(sqrt least-positive-single-float)** used to give 3.7454153e-23, but should be 3.743392e-23.

Results were somewhat worse for double-precision: **(sqrt least-positive-double-float)** used to give 2.6369216085867632d-155, but should be 2.2227587494850775d-162.

Note that in Genera 7.2, normalized arguments to **zl:sqrt** and **sqrt** functions gave results correctly rounded to all significant places, as specified by the IEEE standard. In Genera 8.0, that is now true of all arguments, both normalized and denormalized.

Floating-Point Numbers Work More Reliably

ash and **scale-float** of floating-point numbers did not always interact correctly with **without-floating-underflow-traps**. Other functions, such as **exp**, which use **scale-float** to produce their results, were also affected. Now, for example,

```
(without-floating-underflow-traps (exp -1e20)) => 0.0
```

without signalling underflow.

log Handles Bignums and Large Ratios Correctly

log now correctly handles rational arguments that are too large to be directly converted to floating-point. For example:

```
(log (expt 10 100)) => 230.25851
(log (expt 10 100) 10) => 100.0
(log (expt 10 1000) 2.0d0) => 3321.928094887362d0
```

floor Handles Floating-Point Quotient Underflow Correctly

floor now correctly handles cases where a floating-point quotient underflows to 0.

For example,

```
(floor 1.0e-44 100)
```

returns two values:

```
0, 1.0e-44
```

Change to Internal Arguments in arglist

Users who design advice for functions that accept hidden arguments (such as methods and lexical closures) will notice that the number and nature of the hidden arguments to those functions have changed for Ivory machines. The nature of such arglists is now machine-dependent.

For more information, see the section "Designing the Advice".

Changes to load

The **:default-package** keyword has been added to **load**. This keyword specifies the package to be used if the file's attribute list does not specify a package.

load binds not only the value of ***package***, but also the standard value of ***package***. If **in-package** is executed while a file is loading, it sets not only the value but also the standard value of ***package***. This heuristic technique is intended to ensure that ***package*** is not gratuitously rebound by the Debugger if an error occurs while a file is loading. This was a Genera 7.2 ECO.

The **:reprompt** keyword has been removed. It is replaced with a new possible value for the **:if-does-not-exist** keyword. The **:reprompt** value for **:if-does-not-exist** reprompts instead of signalling.

Changes to the Virtual Memory System

A new package, **storage**, contains the virtual memory system implementation, disk drivers, and so on. All the documented interface names, and some interfaces that remain undocumented, are accessible through the **system-internals** package as before. For example, **si:wire-words** is read as the symbol **storage:wire-words**.

If a program used an undocumented interface to the virtual memory system that no longer exists in the **si** package, it might have been moved to the **storage** package. However, below the documented interfaces there are significant incompatibilities between the Ivory and 3600-family storage systems, so the old usage may not be valid or machine-independent.

Change to the Infix Reader

The infix reader (#◊...◊) is now sensitive to the readtable in trying to decide what "/" denotes. In Common Lisp syntax, infix "/" does rational division as **cl:/** would do. In Zetalisp syntax, "/" does truncating division, as **zl:/** would do. The default is Common-Lisp if the syntax cannot be determined.

Improvements in the Ivory Compiler

- The Ivory architecture allows an unlimited number of constants in a compiled function.

- The compiler has been modified to allow most functions with a large number of local variables to compile. The compiler does this by internally splitting the function into smaller functions which can be supported by the architecture. In Genera 7.2, if an array-register declaration was used with a variable that needed to be accessed through an environment (heap or stack-consed), the compiler would give an error. The compiler now gives a warning and uses a normal (as opposed to a fast) array reference in those situations. For example:

```
(defun foo (a b)
  (let ((a a)
        (b b))
    (declare (sys:array-register a b))
    (values (aref a 0)                          ;See note 1 below
            (aref b 0)                          ;See note 2 below
            (bar #'(lambda () (aref a 0))))))   ;See note 3 below
```

```
For Function (:INTERNAL FOO 0)
  FAST-AREF of A being turned into AREF
For Function FOO
  ARRAY-REGISTER declaration for A being ignored
  FAST-AREF of A being turned into AREF
```

Notes on the example:

1. This uses a normal array reference, since A is part of a heap-consed environment. The compiler used to give an error, but will now generate a warning.

2. This will use a fast array reference.

3. Same as note 1.

Here is another example:

```
(defun foo (a b)
  (let ((a a)
        (b b))
    (declare (sys:array-register a b))
    (values (aref a 0)                           ;See note 1 below
            (aref b 0)                           ;See note 2 below
            (bar #'(lambda ()
                     (declare (sys:downward-function))
                     (aref a 0))))))             ;See note 3 below
```

```
For Function (:INTERNAL FOO 0)
  FAST-AREF of A being turned into AREF
```

Notes on the example:

1. This will use a fast array reference. Even though A is part of an environment, the environment is stack-consed and shared with the current stack frame; A can be accessed in the current frame.

2. This will use a fast array reference.

3. This will use a normal array reference. Even though A is in a stack-consed environment, the environment is located in a different stack frame from the one in which the reference occurs, so A must be accessed through the environment.

Miscellaneous Changes

- Body declarations are now allowed in the **do-symbols** macros.

- **sys:region-valid-p** has been moved to the **sys** package from the **si** package.

- **tyipeek** now works for characters outside the Symbolics character set.

- A new function has been added. See the function **sys:%p-store-type-and-pointer**.

- A new Ivory compiler variable has been added. See the variable **compiler:*enable-frame-splitting***.

- A bug in **si:fix-fep-file-internal** that caused an error has been fixed. This was a Genera 7.2 ECO.

- Wiring pages using **si:wire-pages** did not always succeed; the symptom was often a "Data is not wired" error from the user-disk code. This problem has been fixed. This was a Genera 7.2 ECO.

- Printing locatives has been made more robust.

- Printing rational numbers no longer fails when printing certain ratios, such as those under **base :roman**.

- Printing integers now does better error checking of ***print-base***.

- Underflow is no longer signaled when a denormalized remainder is being computed.

- Reexecuting an **advise-within** of an internal function no longer results in an error.

- Array functions such as **sys:array-element-size** and **sys:array-displaced-p** now check that their argument is an array before accessing it.

- **sqrt** of double-precision numbers is now correctly rounded. **sqrt** also now handles the inexact result trap correctly.

- **zl:sqrt** now correctly handles signalling errors for negative arguments.

- Coercions are now handled correctly when **log** is called with two arguments.

- The performance of three mathematical functions (**math:multiply-matrices**, **math:transpose-matrix**, and **math:invert-matrix**) has been improved significantly.

- A bug was fixed in which **setf** of **nthcdr** did not expand correctly.

- Inline functions with keyword arguments no longer fail when the Common Lisp feature of naming the parameter variable explicitly is used.

- Incorrect coercion of values returned by **min** and **max** in certain cases on Ivory-based machines has been fixed to match the 3600-family implementation. The first example below illustrates the previous result on Ivory machines; the second example shows the result on both 3600s and Ivory machines now.

  Incorrect: `(min 3 4.0d0) => 3.0d0`

  Correct: `(min 3 4.0d0) => 3`

- The printed representation of many arrays has been changed to be more readable and understandable to Common Lisp programmers. This change applies when ***print-array*** is **nil**, or the **:array** option to **write** is **nil**. (These are both the default settings.) The printed representation of arrays now includes the specific type of object that can be stored into the array (if it is not **t**), as well as information on the leader and fill-pointer, and whether it is indirect, displaced, stack-allocated, or forwarded.

Examples:

```
#<ARRAY 5x5 simple 101242212>
#<ARRAY BIT 557x757 conformally-indirect 20123644671>
#<ARRAY BOOLEAN 89 fill-pointer 4 101306615>
```

- The printed representation of compiled functions now includes information on whether the function is the current definition, encapsulated, a special form, or macro. The printed representation of lexical closures now distinguishes stack-consed lexical closures from heap-consed lexical closures, to aid in debugging.

- Compiler warnings for variables bound but never referenced are now issued even if the variable is set, but never read. Some programs that used to compile without warnings will now get warnings. For example:

```
(defun foo (x) (setq x 3) 4)
For Function FOO
  The value of variable X was never used.
```

New Features in the Lisp Language in Genera 8.0

CLOS is Available in Genera 8.0

Genera 8.0 provides an implementation of the Common Lisp Object System (CLOS). Symbolics CLOS supports Chapters 1 and 2 of the CLOS specification, the part that has been accepted by the X3J13 Committee, which is specifying ANSI Common Lisp. Symbolics CLOS is an optimized implementation of CLOS, which is suitable for both the development and delivery of CLOS applications.

It is necessary to recompile CLOS code developed with the Beta Test CLOS for use in Genera 8.0. No source changes are necessary for programs which use only the documented interfaces to CLOS.

For information on system tools for converting from Flavors to CLOS, see the section "Enhanced Conversion Tools in Genera 8.0".

Overview of the CLOS Documentation

Symbolics CLOS comes with the following documentation:

Object-Oriented Programming in COMMON LISP

This book, by Sonya E. Keene, presents CLOS and techniques of object-oriented programming in a tutorial style. It is useful for Lisp programmers who are new to CLOS, or who are new to object-oriented techniques. This book contains many examples of programming with CLOS. This book is not available online via Document Examiner.

"Overview of CLOS"

This section introduces the concepts of CLOS and shows a simple example of using CLOS. See the section "Overview of CLOS".

"Symbolics CLOS"

This is the reference documentation for the Symbolics implementation of CLOS. It documents the important mechanisms of CLOS, and every CLOS operator (function, macro, and so on) in a reference style. It does not provide a tutorial approach or many examples, because that approach is taken by the book *Object-Oriented Programming in COMMON LISP*. For Genera users, the reference documentation is available online via Document Examiner.

This documentation is based on information covered in the "Common Lisp Object System Specification", which is the complete definition of the behavior of CLOS and which should be considered the primary source of information on CLOS until the ANSI specification of Common Lisp itself is completed and available. The authors of the "Common Lisp Object System Specification" (Daniel G. Bobrow, Linda G. DeMichiel, Richard P. Gabriel, Sonya E. Keene, Gregor Kiczales, and David A. Moon) have given Symbolics their permission to include and/or adapt information from the specification into this reference documentation.

The CLOS specification is not provided with Symbolics CLOS. If you are interested in the CLOS specification, its reference is:

> "Common Lisp Object System Specification," X3J13 Document 88-002R, June 1988. Daniel G. Bobrow, Linda G. DeMichiel, Richard P. Gabriel, Sonya E. Keene, Gregor Kiczales, and David A. Moon.

The CLOS specification is available in:

> SIGPlan Notices (ISSN 0362-1340)
> Volume 23
> Special Issue -- September 1988
> Copyright 1988 by the Association for Computing Machinery.
> ISBN 0-89791-289-6
>
> Price: Members $10.50, Nonmembers $14.00.
> ACM Order Number: 548883
>
> Additional copies may be ordered prepaid from:
> ACM Order Department
> P.O. Box 64145
> Baltimore, MD 21264

Integration Between CLOS and Flavors

Mixed use of Flavors and CLOS is not supported at present. That is, a class cannot inherit from a flavor; a flavor cannot inherit from a class; CLOS generic functions cannot be used with Flavors methods and instances; and Flavors generic functions cannot be used with CLOS methods and instances.

Therefore if you convert a program to CLOS, you must convert all uses of a given flavor to use a class instead, and all Flavors methods for a given generic function to be CLOS methods instead.

Genera offers an automatic tool for converting Flavors programs to CLOS. CLOE does not have such a tool. Genera users only: see the section "Flavors to CLOS Conversion".

CLOS Packages

The **clos** package contains CLOS symbols. There are two categories of symbols in the **clos** package:

- Symbols which are included as a standard part of the CLOS specification (this includes Chapters 1 and 2 of the specification, which is the programmer interface to CLOS). These symbols are exported from the **clos** package, documented, and fully supported by Symbolics.

- Symbols which implement the stable and portable portions of the meta-object protocol. We believe that these symbols will be present in any future meta-object protocol standard. These symbols are exported from the **clos** package and can be used by users, although they are neither documented nor fully supported. These symbols are not defined in the CLOS specification, but many of them are present in compatible form in other implementations of CLOS. All are compatible in Symbolics Cloe. These symbols might change in a future release to reflect ongoing standardization efforts.

We recommend that users do not use the internal and external symbols of the **clos-internals** package, which implement the portions of the meta-object protocol which we believe are not defined well enough to be standardized. A future meta-object protocol might provide similar functionality in a different way. These symbols are not exported from **clos**, not documented, and not supported.

The **future-common-lisp** and **future-common-lisp-user** packages contain an incomplete, unsupported implementation of the future ANSI Common Lisp standard. The **future-common-lisp** package contains symbols in the evolving ANSI Common Lisp standard. It is the ANSI Common Lisp equivalent of the **cl** package, although many of the functions have not yet been modified for ANSI Common Lisp, and many are not yet defined. Since ANSI Common Lisp includes CLOS, **future-common-lisp** contains the symbols in the CLOS package that are part of the standard. The **future-common-lisp-user** package is the ANSI Common Lisp equivalent of the **user** package, except that **future-common-lisp-user** uses only the **future-common-lisp** package; it does not include any Symbolics extensions to the language.

If a symbol in the **future-common-lisp** package is documented, then it is supported. We do not encourage users to depend on any undocumented symbols in the **future-common-lisp** package.

We chose the home package of a symbol according to whether the symbol was part of CLOS (**clos** is the home package) or something used in connection with CLOS (**future-common-lisp** is the home package).

Differences Between ANSI CLOS and Symbolics CLOS

This section describes how the current Symbolics CLOS implementation varies from the CLOS specification, which is part of the draft ANSI specification for Common Lisp. The differences fall into two categories: Symbolics extensions to CLOS, and CLOS features or functions that Symbolics does not support.

Symbolics Extensions to CLOS

- The **:locator** slot option to **clos:defclass** is a Genera extension, but CLOE does not support it.

- The function **locf** can be used with **clos:slot-value** to get a locative to a slot. This is a Genera extension, but CLOE does not support it.

- Many of the "Metaobject Protocol" generic functions are implemented to some extent. Although undocumented and subject to change, some of these, particularly the ones exported from the **clos** package, may be of interest.

Incompatibilities with the CLOS Specification

- Classes for the types **function**, **pathname**, and **stream** are not supported.

- The following operators are not implemented: **clos:generic-labels**, **clos:generic-flet**, **clos:generic-function**, and **clos:with-added-methods**.

- Methods that require a lexical environment are not yet supported.

- **clos:describe-object** returns the object being described. This is a deliberate incompatibility.

- When using **clos:ensure-generic-function**, if the **:lambda-list** is not provided, the lambda list of the generic function will not be changed. This reflects a change that will be made to the CLOS specification.

- **clos:symbol-macrolet** does not allow declarations.

- Note that the definition of **clos:symbol-macrolet** has been changed by X3J13 since the CLOS specification was published; the macro expansions are in the environment of the **clos:symbol-macrolet**.

CLOS Incompatibilities with Previous Versions of Genera

&rest arguments have dynamic extent.

CLOS shares the data-type **sys:dtp-lexical-closure** with objects of type **sys:lexical-closure**. Programs that use the predicate **sys:lexical-closure-p** or tested for the Zetalisp type **:lexical-closure** or the Common Lisp type **sys:lexical-closure** should be recompiled. Prior to Genera 8.0, these callers were not recorded, so the command Find String may be the most effective way to find the code that needs recompilation. If this code is not recompiled, CLOS generic functions will be treated as lexical closures. Any programs that manually construct their own lexical closures without using the undocumented function **sys:make-lexical-closure** should ensure that they set the cdr-codes of the two words the same way that **sys:make-lexical-closure** does.

There is a new **setf** symbol, **future-common-lisp:setf**. The macro **future-common-lisp:setf** is no different from the macro **setf**, but, when used in a function name, such as **(future-common-lisp:setf** *symbol***)**, the new value will be the first argument to the function. (This is in contrast to **(setf** *symbol***)**, in which the new value is the last argument to the function.)

In the past, using **setf** on an accessor *symbol* that had no defined setf method caused an error. It now generates a call to the function **(future-common-lisp:setf** *symbol***)**, to conform with ANSI Common Lisp. For additional information, see the macro **future-common-lisp:setf**.

Known Problems with CLOS in Genera 8.0

- The **:arguments** option to **clos:define-method-combination** is not implemented. This problem will be fixed in an ECO or future release.

- In methods, using **eql** specializers for objects where **eql** is not **eq** is not implemented. This means **eql** specializers do not work for objects of the classes **bignum**, **complex**, **double-float**, **ratio** (3600 only), and **sys:big-ratio** (Ivory only). This problem will be fixed in an ECO or future release.

- When a method combination type is redefined, generic functions that use that method combination type will not be updated. This problem will be fixed in an ECO or future release.

- Combined methods appear as the function **nil** in the Debugger.

- Encapsulations are not yet supported. Thus, methods and generic functions cannot be traced, advised, and so on.

- The linker has not been extended to CLOS. When this happens, all generic function calls will be faster on Ivory machines.

- **future-common-lisp:defstruct** does not work interpreted on Ivory. This problem will be fixed in an ECO or future release.

- In some cases **(setf slot-value)** is not optimized. This problem will be fixed in an ECO or future release.

- Redefinition of classes may cause some problems with multi-methods, particularly on 3600s. This problem will be fixed in an ECO or future release.

Common Lisp Developer is Available in Genera 8.0

Genera 8.0 provides the Common Lisp Developer, a tool that assists in developing and testing Common Lisp programs. It is useful when you are planning to port a program to other Common Lisp implementations.

The Common Lisp Developer corresponds to Common Lisp as it is described in the 1984 edition of Guy Steele’s *Common LISP, the Language*. A second edition of the book has recently appeared, which is a snapshot of work in progress by an ANSI committee that is trying to produce a standard for Common Lisp. The Common Lisp Developer does not currently attempt to support the details of that emerging standard.

For more information, see the section "Developing Portable Common Lisp Programs".

New Function: string-a-or-an

string-a-or-an *string* &optional *(both-words* **t***) (case* **:downcase***)* *Function*

Computes whether the article "a" or "an" is used when introducing a noun. If *both-words* is true, the result is the concatenation of the article, a space, and the *noun*; otherwise, the article is returned. The *case* argument controls the case of the article. For example:

```
(string-a-or-an ’rock)          => "a ROCK"

(string-a-or-an ’rock t :upcase) => "A ROCK"

(string-a-or-an "egg")   => "an egg"
```

New format Directive: ~\\QUOTED-STRING\\

Prints a quoted string. The modifiers work as follows: A colon prints the quotes only if they are needed for the current readtable, and an at-sign uses a vertical bar (|) instead of a double-quote (") for quoting. For example:

```
(format t "~{~%~\\quoted-string\\ ~:\\quoted-string\\ ~
            ~@\\quoted-string\\ ~:@\\quoted-string\\~}~%"
           '("foo" "foo" "foo" "foo" "foo bar" "foo bar" "foo bar" "foo bar"))
"foo" foo |foo| foo
"foo bar" "foo bar" |foo bar| |foo bar|
```

New Function: future-common-lisp:fmakunbound

Genera 8.0 offers the new function **future-common-lisp:fmakunbound**, which is just like **fundefine**.

See the function **future-common-lisp:fmakunbound**.

setf of fdefinition in Genera 8.0

In Genera 8.0, you can use **setf** with **fdefinition**.

New Option to make-hash-table

Genera 8.0 offers the **:gc-protect-values** keyword to **make-hash-table**. The default is **t**. If **nil**, table entries are automatically deleted if a value becomes unreachable other than through the table. This keyword is a Symbolics extension to Common Lisp.

Improvements to the Lisp Language in Genera 8.0

Improvements to the Compiler in Genera 8.0

- The Ivory compiler generates considerably more efficient code than in Genera 7.4 Ivory, taking advantage of the full Ivory instruction set.

  The 3600-family compiler is more efficient than in Genera 7.2, but the improvement is not as dramatic as that of the Ivory compiler.

- All environment-free functions are now compiled by the file compiler.

Package DWIM is Faster in Genera 8.0

The "package DWIM" facility is many times faster in this release. You may want to remove any local customizations that disable it.

Improvements in Compiler Handling of import and export At Top Level

- **import** and **export** now signal errors if the correct package is not specified. Given a symbol **S** with home package **H**, and given current package **C**, the compiler's top-level form handler for **import** used to rewrite

  ```
  (import 'h:s)
  ```

  as

  ```
  (import (intern "s" (find-package "h")))
  ```

  If "H" was not found at load time,

  ```
  (find-package "h")
  ```

  returned **nil** and the **intern** was completed in the current package rather than in package H. You would then get the erroneous effect of

  ```
  (import 'c:s)
  ```

  An error is now signalled instead of using the wrong package.

  Also, given the same configuration of symbols, **export** would rewrite

  ```
  (export 'h:s)
  ```

  as

  ```
  (export (intern "s" (find-package "h")))
  ```

  or

  ```
  (export (intern "s" *package*))
  ```

  depending on whether "S" was "already present". If the same file was compiled twice in the same environment, the symbol was always present on the second attempt, so the second rewrite style was used in that case — sometimes erroneously. An error message is now repeated with each compile.

- The following bug has been fixed in Genera 8.0: **export** was rewriting

  ```
  (export '(a b ...))
  ```

  as

  ```
  (progn (export (intern "a" ...)) (export (intern "b" ...)) ...).
  ```

  This was incorrect in the case of

  ```
  (export '(nil))
  ```

  which is not the same as

  ```
  (progn (export (intern "nil" ...)) ...)
  ```

  since the result of the **intern**, if it is **nil**, is treated as an empty list. **nil** now exports correctly.

Miscellaneous Improvements to the Lisp Language in Genera 8.0

- The arglist for **return-from** has been corrected to be *block-name* &optional *values*.

- To correct a bug that existed in Genera 7.3 Ivory and Genera 7.4 Ivory, you must recompile code that uses any of these functions: **char>**, **char≥**, **char<**, and **char≤**. Recompiling your code will not damage previously working code.

- A performance improvement has been made by decreasing the amount of paging necessary for resources. In order to take advantage of this performance improvement, you must recompile existing code that uses **allocate-resource**, **using-resource**, and **deallocate-resource**.

- Multidimensional stack arrays now work with the **:initial-contents** option.

- **setf** and **locf** now correctly handle functions defined by **defsubst-in-flavor**.

- On Ivory-based machines, compiled function calls where the last argument was the result of one of many floating-point arithmetic operations work now. If you use floating-point, you should recompile your code to take advantage of this fix.

- The default for the **defstruct** option :**eval-when** is now (**load eval**).

- **deflambda-macro** has been extended to permit the use of &environment.

- **string-pluralize** now correctly pluralizes the words "human", "child", and variants of "child". For example:

  ```
  (SCL:STRING-PLURALIZE "grandchild") => "grandchildren"
  ```

- The following bugs have been fixed:

  ° `(directory-namestring "S:foo")`

    used to return **nil**, but now returns "".

  ° `(documentation nil 'variable)`

    used to signal an error, but now returns **nil**.

- New optional arguments are available for the function **last**:

  ° `(last` *x* `1)` returns the last cons of the list *x*.

  ° `(last` *x* `0)` returns the **cdr** of the last cons of the list, and is useful for dotted lists. For example:

  ```
  (last '(A B . C) 0) => C
  ```

  ° `(last` *x n*`)` returns the list of the last *n* conses of the list.

Changes to Utilities in Genera 8.0

Changes to Utilities Since Genera 7.2

New Scheduler Available for 3600-Family Machines

The new scheduler introduced in Genera 7.3 Ivory is present in Genera 8.0. There are two important changes in the way the scheduler works:

- The new scheduler is event-driven; the old scheduler polled.

- The new scheduler does not have a separate stack-group; the old scheduler did.

This means that you can write programs that use processes more efficiently, which improves performance.

For a complete description of the new scheduler, see the section "The Scheduler".

For a discussion of conversion issues, see the section "Converting Code for the New Scheduler".

tv:*select-keys* is Obsolete

The variable **tv:*select-keys***, previously used by the SELECT key, is obsolete. it is retained for compatibility. (The SELECT key now uses an internal database.) Use **tv:add-select-key** where possible.

Improvements to the Debugger

The first five items below were Genera 7.2 ECOs.

- Moving the mouse over certain instances of old Debugger stack frames no longer causes an error.

- Function-specs in the Debugger now allow the current language to contribute, so that unusual things like static functions in Symbolics C work properly with the c-E command.

- The value-cell of a monitored location is now properly set when the user unmonitors the location before proceeding.

- Output from the Show Source Code command is once again mouse-sensitive while in the Debugger.

- Evaluating special variables in the Debugger now works correctly when the Debugger was entered from an interpreted frame.

- Multiple processes no longer get into the cold load stream simultaneously, which used to result in confusing overprinting.

- The Debugger now works normally if any sheet is prepared. Previously, it went into the cold load stream.

- When the Debugger evaluates a form, the variable ***read-default-float-format*** is checked to ensure that it contains a legal value. If not, it is set to its standard value.

- Using `c-E` in the Debugger on a remote console now works correctly; previously, it would incorrectly select an editor on the main console.

Improvements to Hardcopy

The first three items below were Genera 7.2 ECOs.

- Greater precision is now used in PostScript streams, so that graphics and clipping are more accurate.

- The wrapping of characters is now handled better.

- Attachments for **graphics:draw-string-image** are now done in the proper coordinate system.

- The user interface for hardcopying a screen has been improved. Pressing `FUNCTION 0 Q` pops up a menu of options. The options depend on your machine's software and hardware configuration.

- It is now much faster to hardcopy a PostScript file to a PostScript printer.

Improvements to the Metering Interface

New Process Time Output Field

In previous releases, the Process Time output field was available only for the statistical types of metering. The Function Call and Call Tree metering types had only the Time output field. The Process Time output field is now available for Function Call and Call Tree metering.

The Process Time output field (whether Inclusive or Exclusive) includes only the time during which the process of interest was running. The Time output field (whether Inclusive or Exclusive) includes all the time spent during the metering run; that is, both the time when the process of interest was running, and the time when other processes were running. You can think of the Time output field as the result of using a stopwatch or wall clock to time your code.

The Process Time output field is usually more valuable information than the Time output field. For this reason, the Metering Interface now displays the Process Time output field by default, replacing the Time field.

New Error Information

New error output fields provide information on the reliability of the metering results. For more information, see the section "Error Output Subfields in Metering Results".

New RAvg Output Subfield

Many output fields now include a new output subfield called RAvg. RAvg means "Reasonable Average", which is the average of samples that fell in the main node of the histogram, only. The partitioning of a histogram into nodes is done heuristically, at runtime, and is therefore imperfect, at best. RAvg is a good first cut at filtering out noise, but closer inspection of the full contents of the histogram is sometimes necessary, if RAvg depends on false samples.

Improved Documentation on Set Display Options

See the section "Setting the Display Options".

Improvement to Statistical Program Counter Metering

Statistical Program Counter metering (also called PC metering) is now more accurate when it is focusing on peaks. This was a Genera 7.2 ECO.

:Without Interrupts Metering Keyword Uses process:without-preemption

The :Without Interrupts keyword is accepted by Function Call, Call Tree, Page Fault, and Statistical Program Counter metering types when the Meter Form command is used. In previous releases, this keyword caused the form to be executed inside a **without-interrupts** form. Now this keyword causes the form to be executed inside a **process:without-preemption** form.

Improvements to the FEP-Tape System's User Interface

- The screen layout has been redesigned for more convenient use.

- The name of the command "Add Files For Microcode Version" has been changed to "Add Microcode Files". (The previous name was easily confused with "Add Files".)

- The file display pane now has a scroll bar and a typeout window, and supports the standard scrolling command accelerators (SCROLL, m-SCROLL, m->, m-<, and so on).

- The mouse sensitivity of the file display pane is now correct with respect to the Remove File command.

Improvements to the System Construction Tool

The first three items below were Genera 7.2 ECOs.

- SCT now uses a more powerful way to determine whether to "compile" files that do not produce any output (such as C "include" files).
- Users who edit the herald or otherwise introduce newlines into the system information are now protected from creating unreadable patch files.
- **sct:set-system-source-file** and **sct:set-system-directory-file** now merge against a logical pathname rather than a physical pathname, so that operating-system-dependent problems do not arise.
- **sct:copy-system** has been reimplemented.
- Logical patch file pathname translation is now faster.
- SCT journalling is now generic, to support multiple machine types.
- In the past, if you marked a version of a system as **:broken** or **:obsolete**, and you tried to open a new patch on it, the Zmacs/SCT patch facility allowed you do it without warning. Now you are queried about whether you really want to patch that system.
- SCT no longer errs when you try to "load" a nonloadable file (such as a .text file) when the file is not there.
- You can now make the *major-version* argument to **sct:designate-system-version** be **nil**, which removes *designator*. For more information: See the function **sct:designate-system-version**.

Improvements to Tape

These were Genera 7.2 ECOs.

- In the Distribute Systems frame, the Write Distribution command for multi-reel plans was just rewinding at the end of a tape, and continuing to write the next reel's data on the same tape. The command now works correctly.
- The Tape system now provides better support for incremental distributions, so that ECOs can be distributed.

- **tape:carry-dump** now screens out directories and links to directories to avoid getting errors.

Improvements to TELNET

These were Genera 7.2 ECOs.

- It no longer echoes some very simple commands.

- It now tracks the number of lines on Ambassador-style terminals.

- It now better detects UNIX protocol violations on bare CR characters.

- It interacts better with Debugger typeout windows.

- Output recording in TELNET windows can now be disabled.

- It now synchronizes the output process against changing NVT options.

- Some **:fast-filter-test** methods have been added to speed up some common UNIX cases, notably, redisplay in UNIX EMACS.

New Initialization List Replaces Some Uses of :warm Initializations

A new initialization list, **:window**, is now available through **add-initialization**. Some things that used to be done with **:warm** initializations should now be done with **:window** initializations, since in embedded systems the window system is not yet running while **:warm** initializations run. (In embedded systems, the Genera window system runs only when the host asks for it.)

For more information, see the function **add-initialization** and see the section "System Initialization Lists".

Metering Interface Works for non-Lisp Languages

You can use the Metering Interface to meter "foreign" programs (such as C, Fortran, or Pascal programs) in the same way you use it to meter Lisp programs. In Genera 8.0, the Metering Interface displays the names of the foreign functions in a language specific format, rather than simply using the internal Lisp function name. This should make the output more understandable to the programmer.

New Command: Show System Version Designations

A new command, Show System Version Designations, allows you to determine designations (such as Released or Latest) of systems that are not loaded in your world. For more information, see the section "Show System Version Designations Command".

New Value to :Use Disk Option to Distribute Systems

If you use the Distribute Systems command on a MacIvory, there is a new value for the :Use Disk command.

On a MacIvory, the choices for :Use Disk are {Tape, Disk, Floppy}. Disk indicates the hard disk, and Floppy indicates the floppy disk. These two values have the same effect as the test mode described below. Tape means to write to tape; this is the default.

On all machines other than a MacIvory, the choices are {Yes, No}. If Yes, the input is written to disk (test mode), rather than to tape. Test mode writes a special file which is an image of what would be written to tape. Use this when you are preparing a distribution and want to see what files would be written to tape. The default is No. The mentioned default is Yes.

For more information, see the section "Distribute Systems Command".

Miscellaneous Changes

The first nine items below were Genera 7.2 ECOs.

- Insertion and removal of topics into the middle of a Document Examiner viewer has been greatly speeded up.

- Encapsulations now work with lexical closures (for the Metering Tools).

- Previously, when the transporter discovered a storage-convention violation, it issued an error message that incorrectly blamed the garbage collector. The message has been rewritten to be more accurate.

  Old message: "Internal error in the garbage collector"

  New message: "A storage-convention violation has been detected by the garbage collector".

- Delayed compiler warnings are now spliced out correctly after they have been printed, so that they do not remain indefinitely.

- A fencepost error in the 16-bit string loader has been fixed.

- **random-state** objects can now be dumped.

- The general resource allocator is now much faster when the resource is large.

- **map-resource** now works correctly when there is a **deallocate-resource** in the continuation.

- Some parts of the reader — **tyipeek** and **set-dispatch-macro-character** — now work better against nonstandard character sets.

- Genera now includes the VT100 font file, which had not been distributed in earlier releases. This file is used by Ann Arbor and VT100-emulator terminals.

- Peek's Processes display includes extensive information about timers. By default, the information is not shown; to display it, click on `Timers`.

- The Save World command now sets the estimated length-in-bytes of the world file before starting to save and the real length-in-bytes before trying to do any console output. This means that the world will be (potentially) bootable if **zl:disk-save** fails after completing the actual saving but before finishing the final dialogue with the user.

- In Genera 7.2, the [Selective Open] menu item operation did not work on directories in FSEdit. This has been fixed.

- Several bugs in the disassembler have been fixed.

- A bug has been fixed in the arglist-finder, which is part of the implementation of `c-sh-A`. The bug caused anything with a null arglist to be treated as undefined. Joshua users would notice this bug when trying to use `c-sh-A` on zero-argument Joshua predicates.

- A bug has been fixed in loading patches. Previously, if you used the Highest query option in loading patches, and you gave the current patch as the highest number to load, then nothing was loaded.

- When you use **(load-patches :silent t)**, the output from **sct:dangerous-patch** warning you that loading of patches for that system has stopped is now displayed; this warning was suppressed in a previous release.

- Finish Patch now allows you to specify a patch-level dependency in another system for a major version number higher than the one loaded. This is a reasonable thing to do; for example, you might be naming a new system version that you know exists, but which you have not yet loaded.

- A bug has been fixed in reloading systems. Previously, if you reloaded a system, no patches for that system were reloaded unless by chance a new patch had been made for that system.

- A bug has been fixed in which calling **ed** or using `c-E` in the Debugger on a remote console would incorrectly select an editor on the main console.

- The bug-report banner in bug mail now includes the name of the world-load file from which the present world was booted.

  On a MacIvory, the banner now includes information on all Macintosh software and hardware (Macintosh II or Macintosh IIx, keyboard type, and so on) in use.

  On a Symbolics UX-family machine, the banner includes the UNIX version and kernel name, the X Server version and vendor name, and the UX Life Support names.

- Multireel distribution tapes that have a large number of files on the tape and a large number of systems in the distribution (such as the Symbolics source tapes) previously got an end-of-tape error while being written by the distribution dumper. This bug has been fixed.

- The following command now works correctly:

  ```
  Compile System <system> :Condition Always :Query <answer>
  Source-Compare
  ```

  Previously, the command would fail trying to print any differences encountered.

- You can now use the Show Machine Configuration command on a 3600-family machine to show the configuration of an Ivory-based machine.

New Features in Utilities in Genera 8.0

Enhanced Conversion Tools in Genera 8.0

Genera 8.0 includes some improvements to the Conversion Tools that were first released with Genera 7.1. These tools are now more powerful and less tedious to use. In addition to Zetalisp to Common Lisp conversion, you can now perform Flavors to CLOS conversion, and you can easily write your own conversions.

For information on the conversion tools, see the section "Conversion Tools".

Hardcopy File Command Has Two New Keywords

Two keywords, :Notify and :Print Cover Page, have been added to the Hardcopy File command. (This was a Genera 7.2 ECO that was not noted in Genera 7.3 Ivory or Genera 7.4 Ivory documentation.) These keywords allow you to suppress the printing of cover pages and the sending of notifications when jobs have finished.

For more information, see the section "Hardcopy File Command".

New Garbage Collection Mode: In-Place Garbage Collection

Genera 8.0 introduces a new mode of Garbage Collection: *In-Place Garbage Collection*.

See the section "In-Place Garbage Collection".

You specify In-Place Garbage Collection via the :Immediately keyword to the Start GC command or via the function **si:gc-in-place**.

See the section "Start GC Command".

Formatting QIC-100 Tapes

You can now format a QIC-100 tape on your SCSI drive. The drive must be local. The command is Format Local SCSI Tape. If you have more than one tape drive on your machine, Format Local SCSI Tape pops up a menu that allows you to specify which drive to use. See the section "Format Local SCSI Tape Command".

You can also purchase preformatted Gammamat 3M DC2000 mini-data cartridge (QIC-100) tapes for use with XL400 and MacIvory systems with QIC-100 tape drives.

New Special Form: compiler:invisible-references

There is a new special form, **compiler:invisible-references**. It allows references to variables in a form to remain invisible to the compiler for the purposes of variable reference warnings.

See the special form **compiler:invisible-references**.

New defsystem :module Keyword Option

A new **defsystem** module option, **:load-when-systems-loaded**, controls the loading of modules according to whether or not required systems have been loaded.

See the section "**:module** Keyword Options".

New Command: Verify Distribution

A new Command Processor command, Verify Distribution, reads a Distribution from tape, floppy, or disk, and compares the contents of each file in the distribution to the corresponding file in the file computer.

See the section "Verify Distribution Command".

New Function: si:define-application-logout-initialization

A new function, **si:define-application-logout-initialization**, lets you define an application logout command to be used by an application to execute a form before the user logs out. For example, Zmacs uses this function to save modified editor buffers.

See the function **si:define-application-logout-initialization**. (Note that in Genera 8.0 this function is documented online, but not in the printed documentation set.)

New Remote Terminal Commands

You can use three new Command Processor commands from remote terminals in Genera 8.0:

- Set Remote Terminal Options

- Show Remote Terminal Options

- Halt Remote Terminal

These commands are not currently documented.

Improvements to Utilities in Genera 8.0

Improvements to dis:copy-distribution-tape in Genera 8.0

Prior to Genera 8.0, **dis:copy-distribution-tape** only allowed you to copy from tape to tape or from disk pseudotape to tape (but not tape to disk), or to or from a floppy on the MacIvory.

dis:copy-distribution-tape's "control panel" menu now allows a device of Tape, Disk, and (on a MacIvory) Floppy, for both source and destination.

Distribution System Now Supports Nine-Track Cartridge Tape Drives

Distribution writing commands now have the same kind of keyword argument for tape capacity as the FEP-Tape commands. The default for all of them is now the nine-track capacity.

The following commands have the following changes:

Command Processor:

Distribute Systems	New keyword, :Full Length Tapes (default Yes)

Distribute Systems frame:

Write Distribution	New keyword, :Full Length Tapes (default Yes)

Fep-Tape frame:

Write Tape	New default for keyword :Full Length Tapes is Yes
Read Tape	Remove the keyword :Full Length Tapes

Verify Tape	Remove the keyword :Full Length Tapes

Improvements to TAR Tape Handling

A tar-tape writing capability now exists. Two new Command Processor commands, Write TAR Tape and Read TAR Tape, provide the user interface.

The tar-tape reading capability has been improved. Files containing styled characters that are written to a UNIX, then dumped to a tape, now read back onto Genera correctly.

The old code (which could only read tar tapes) is still available in the file SYS:UNSUPPORTED;READ-TAR-TAPE.LISP.

Improvements to SUPDUP in Genera 8.0

- When typing to a SUPDUP connection in a Terminal window, you can now press c-SCROLL to send code 37 octal (sometimes called c-_ by remote hosts).

  (Before the Terminal window became a Dynamic Window, pressing SCROLL used to send this character. When the Terminal window was made dynamic, the Scroll key was redefined to scroll the window, losing the ability to send that code in a convenient way.)

- SUPDUP connections normally encode characters modified by CONTROL-, META-, SUPER-, or HYPER-, such as CONTROL-A, as "A" with a Control- bit set, and CONTROL-SHIFT-A as "a" with the Control- bit set. This is normally the right thing. However, some remote hosts or remote host programs expect to see the opposite. If you are working with such a host or program, where CONTROL- and CONTROL-SHIFT- characters are interpreted backwards, you can set the variable **telnet:*itp-shift-bucky-chars***, which defaults to **t**, to **nil**.

Miscellaneous Improvements

- The SCT command Compile System *system* :Version *version* works.

- Patch atoms can now be sensitive to case in UNIX systems; if you explicitly supply a **:patch-atom** in a **defsystem**, it should be in interchange case. (For more information on interchange case, see the section "Case in Pathnames".)

- A bug prevented system modules in SCT from accepting a **:version** clause to specify precisely which version of the component system to load. This bug has been fixed.

- LGP3 printers are now fully supported.

- Printer control commands — Reset Printer, Restart Printer Request, and Delete Printer Request — have all been improved for Genera 8.0, and take new arguments.

- Slow band-transfer (used by the Copy World command) is now considerably faster.

- In earlier releases, when the Ivory Debugger displayed &rest (and &key) args, it did so twice, once as individual args, once together as the &rest arg. In addition, the Debugger displayed all &rest args as "extraneous" args. These bugs have been fixed.

- Two unimplemented keywords, **:flavor** and **:unbound-function**, have been removed from **who-calls**. A third keyword, **:microcoded-function**, remains implemented on 3600-family machines only.

- **si:full-gc** now works in delivery worlds.

- The Ivory stepper and breakpoints now work.

Incompatible Changes to Utilities in Genera 8.0

Changes to FSEdit in Genera 8.0

The meaning of the Left and Middle mouse clicks on FSEdit objects have been changed:

Left	Open/Close object. If the object is a directory, it shows or hides the directory contents. If the object is a file, it does Show File on the file.
Middle	Delete/Undelete object.
Right	Menu of operations. See the section "FSEdit Commands".

By default, deletion via this command always requires confirmation. You can set the variable **tv:*confirm-fsedit-quick-soft-file-deletion*** to **nil** if you want to disable the confirmation of *soft* deletion (deletion of a file on a host which supports undeletion). No mechanism is provided for disabling confirmation on a host that does not have soft deletion.

If you want to restore the former behavior, you can set the variable **tv:*use-new-fsedit-command-set*** to **nil**.

For more information, see the section "FSEdit Commands".

Known Problems with Utilities in Genera 8.0

Known Problems with Print and Hardcopy

- The following keywords to the Hardcopy File CP command and to **hardcopy:hardcopy-file**, **hardcopy:hardcopy-text-file**, **hardcopy:hardcopy-from-stream** and **hardcopy:make-hardcopy-stream** do not work:

  :heading-character-style
  :body-character-style

- Making multiple copies with **:orientation landscape** does not work. Only the first copy is printed in landscape orientation.

- When the printer is disconnected or does not respond for more than four minutes, the background process goes into `Serial Unit Locked` state.

Changes to the User Interface in Genera 8.0

Changes to the User Interface Since Genera 7.2

New Defaults for the Logout Command's Keyword Arguments

The Logout command's three keyword arguments — Save Buffers, Save Mail, and Save Spell Dictionaries — have new defaults and simplified options. In addition, there is a new keyword argument, Expunge Mail.

The default for Save Buffers, Save Mail, and Save Spell Dictionaries was previously Yes; it is now Ask. The possible values are:

Yes	Saves modified objects without further prompting.
Ask	Queries whether or not to save each modified object individually. This is the new default.
No	Does not save any modified objects.

The new keyword argument, Expunge Mail, supersedes the Save Mail keyword argument's Expunge-As-Well value. Save Mail now allows only the values Yes, No, and Ask. Expunge Mail has two values, Yes and No.

Yes	Expunges mail file buffers before saving them. This is the default.
No	Does not expunge mail file buffers before saving them.

Changes to the Save Mail Buffers and Save File Buffers Commands

The Query keyword argument to the Save File Buffers and Save Mail Buffers commands now takes only the values Yes (the default) or No. In earlier releases, it accepted any of the values Yes (the default), No, or Ask.

Similarly, the Expunge keyword argument to the Save Mail Buffers command takes the values Yes (the default) or No. Previously, it accepted Yes, No, or Ask (the default).

Improvements to Command Processor Commands

These were Genera 7.2 ECOs.

- The Set Window Options command now works correctly on color screens.
- The Show ECOs command now presents ECOs in a better order, and the Show Herald command displays ECO levels more readably.
- The Show FEP Directory command now handles gaps in disk unit numbers properly.
- The Copy World command no longer fails if the remote world file has extra blocks at the end that do not need to be transferred.
- You can now use the Show Source Code command on traced functions.

New Options to the Set Screen Options Command

The Set Screen Options command has new options to control the repeat functions for the keyboard and the mouse and the position and size of the background interactor window. Graphics Scan Conversion has also been added. The function **tv:set-screen-options** (for use in setting screen options from an init file) has been similarly updated. For more information, see the section "Set Screen Options Command" and see the function **tv:set-screen-options**.

Changes to the Copy Flod Files Command

The Copy Flod Files command now accepts an :Automatic keyword argument. This argument specifies that :Copy Flod Files should automatically skip copying the flod files to any hosts for which the process gets an error. This change allows the Copy Flod Files command to proceed unmonitored. The default setting of this argument is Yes if more than one target host is specified; otherwise, it is No.

The command also now offers a proceed option to skip a host when errors occur. In addition, a bug has been fixed that caused Copy Flod Files to get the same error twice if a target host was not available.

In addition, the Copy Flod Files command now checks for the FEP kernel on Ivory-based machines, and can copy FEP kernels to the local machine only. You must cold boot the FEP after copying, to keep the flods files and the kernel in sync. Ordinarily, you must also cold boot Lisp after copying a new kernel.

Note that on 3600-family machines, the FEP kernel is in hardware read-only memory; on Ivory-based machines, it is on the disk, so it can be changed easily. The disk label includes the disk address of the kernel, used when cold booting the FEP.

Selecting a Group of Objects with the Mouse

You can use the mouse to sweep over an area and select all the mouse-sensitive objects of a given presentation type in that area to use as input to a command that takes a sequence of arguments.

For example, type `Show Directory sys:examples;` .

Type `Show File` and position your mouse at the right hand side of the directory listing. Click and hold the left mouse button and drag the mouse down and to the left. All the pathnames in the rectangle you create by the diagonal motion are added as a sequence to the Show File command when you release the mouse button.

Note that you can mark groups of objects using the click and hold action, but if your mouse is over an object of the same type as the group you want to select (that is, if your mouse is over a single pathname, in this case), that single object is selected. To select a group, your mouse must initially be over a part of the screen that is not mouse sensitive.

New Host Type Accepted by Show Users Command

You can specify the following to the Show Users command, where *embedded-host* is the name of a Symbolics UX400S embedded in a non-Symbolics host:

```
Show Users @embedded-host|HOST
```

For example:

```
Command: Show Users (users [default @RDPCO]) @luxury|host
Login       Name              TTY Idle    When    Where
backup  Tape Drudge            a  4:42 Tue 09:57  Computer room
regis   Regis Shepherd         p0 6:40 Tue 09:20  soleil.rdpco.symbol
```

Miscellaneous Changes

The first seven items below were Genera 7.2 ECOs.

- The Select Key process no longer loses characters when the mouse is in the scroll bar.

- Input Editor state-saving now works correctly when there is no typein.

- Dead windows are no longer claimed to be selectable.

- `c-m-SCROLL` and `c-SCROLL` now work properly when typing commands in Document Examiner.

- A bug that caused the screen dimmer to fail occasionally to brighten the screen after keyboard or mouse activity has been fixed.

- Trying to get help on some topics in the Metering Interface no longer traps to the Debugger when the help topic is a list.

- The status line process no longer gets an error when the current readtable is something nonstandard.

- The Load Patches command has a new keyword, :Excluding, that lets you specify a system or systems for which patches should not be loaded. For more information, see the section "Load Patches Command".

- The Load File command has a new keyword, :Silently, that lets you request notification as each file is being loaded. For more information, see the section "Load File Command".

- The generation of addressees for bug reports has been improved and integrated for all means of creating reports.

- Typing spaces followed by `c-m-T` (Exchange Expressions) to the input editor would cause an error in earlier releases. The input editor now beeps.

- In earlier releases, there was an inconsistency which has been fixed. If you typed a `RETURN` to a Lisp Listener at top level, it would reprompt, giving the appearance of having accepted a "null command". But if you typed `SUSPEND` to get to a break loop and typed `RETURN`, it would be parsed as input, just putting the cursor on the next line; this incorrect behavior has been fixed, and `RETURN` reprompts just as the Lisp Listener does.

- A previous release contained an incompatible change in **login** that prevented use of symbols for the user-name argument, as a side-effect of new code allowing the argument to be an object of class USER. This bug has been fixed.

- In a previous release, when you did **accept** in a non-dynamic window and typed `HELP`, if you tried to use `s-R` or `s-S` to search the Help output, an error was signalled. The current behavior is to beep and return, if the given window has no output history.

- In a previous release, the Show Font command did not work when the typein character style was other than NIL.NIL.NIL, because the font symbol it looked for would have bold characters in its name, but the font symbol that was loaded would not.

- The Add Services To Hosts and Remove Services From Hosts commands now work properly when the Lisp Listener window's typein character style is set to something other than NIL.NIL.NIL.

- Typing a remote tape specification to an **accept** of **'tape:tape-spec** in non-null character style now works properly; in the past, it triggered an error.

- The Help frame (invoked by SELECT HELP, FUNCTION HELP, and related commands), no longer forces the vertical scroll bar into the left margin; user settings can now place it on the right margin.

New Features in the User Interface in Genera 8.0

New Command Processor Commands in Genera 8.0

There are a number of new CP commands in Genera 8.0. Many of them are useful for remote terminal use:

Boot Machine Command
: Boots the machine, using the specified *boot-file*.

Create File Command
: Uses the Simple editor to create a new file or to supersede a file that exists already.

Edit File Command
: Enters the specified editor and reads in *pathname*. As of Genera 8.0, the Simple editor is available for use on small files.

Monitor System Status Command
: Displays a summary of your machine's status.

Read TAR Tape Command
: Reads a tape in TAR format.

Scan Mail Command
: Allows you to look over your new mail, a message at a time, without loading the file into Zmail.

Show Disk Usage Command
: Displays information about the amount disk space used by *files*.

Show Editor Buffer Command
: Displays the contents of the specified editor buffer (*buffer*).

Show Messages Command
: Displays the contents of the specified Converse conversations.

Show Open Files Command
: Shows the files your machine has open for reading or writing and their state.

Show Output History Command
Displays the output history for the specified window.

Show Time Command
Displays the current day, date, and time.

Show Zmail Status Command
Shows the state of your Zmail process.

Write TAR Tape Command
Writes a tape in TAR format.

Improvements to the User Interface in Genera 8.0

Changes to the Copy File Command in Genera 8.0

- The Command Processor command Copy File now accepts sequences of files instead of just a single file. You can click on the displayed names of multiple files for copying rather than having to copy each name individually.

  For example, try marking the output of a Show Directory command using region marking (hold down the Left mouse button starting at the right side of the list of files while in the input of a command that allows sequences of files and drag the mouse over the files you wish to operate on).

  The command is upward compatible, and callers of **si:com-copy-file** with a single file will still work.

- Copy File's :Mode keyword accepts two new values, :Mode Macintosh and :Raw character.

Some Functions Are Now Undoable

Genera 8.0 supports the following newly undoable functions. You can include these functions in **login-forms** in your init files.

- **net:telnet-server-on**
- **tv:setq-who-line-control-variables**
- **net:remote-login-on**
- **zwei:set-variable-value**
- **si:add-ie-command**
- **psetf**
- **si:delete-ie-command**

setf now warns if you specify more than one pair, and suggests alternatives with mechanical undo forms.

The undo function for **cp:cp-on** now calls **cp:cp-off** by default.

Miscellaneous Improvements to the User Interface in Genera 8.0

- A bug in **fs:force-user-to-login** could cause it to prompt for login when the machine was already logged in as Lisp-Machine, sometimes during warm booting. This bug has been fixed.

- The Show Directory command now takes a :Author keyword to limit the directory display to files written by the specified author.

- Numeric arguments to commands invoked by the FUNCTION key (for example, the 0 in FUNCTION 0 S or the 4 in FUNCTION 4 T) are now shown in the status line at the bottom of the screen as you are typing the command.

- The behavior of the Add Paging File command's :Prepend keyword has changed. Specifying :Prepend Yes still forces prepending, but specifying :Prepend No no longer precludes it.

Incompatible Changes to the User Interface in Genera 8.0

Changes to the Time-Setting Commands

The Command Processor commands Initialize Time and Set Calendar Clock have been eliminated. The functionality they provided has been folded into the Set Time command.

Set Time now takes two keywords, :Set Status Line and :Set Calendar Clock. :Set Status Line specifies whether to set the Genera internal clock, which is visible in the status line; this argument defaults to Ask. :Set Calendar Clock specifies whether to set the internal backup clock of either the Symbolics machine or the embedding host computer, as the case may be; it defaults to Ask unless the first argument specifies the source as "Calendar Clock", in which case it defaults to No.

See the section "Set Time Command".

Changes to the Printer Control Commands

The keywords to the Command Processor commands Reset Printer, Delete Printer Request, and Restart Printer Request have been simplified in Genera 8.0.

Known Problems in the User Interface in Genera 8.0

The Command Processor commands Show Timer Queue and Delete Timer Queue are obsolete and will be removed in a future release. If you use them in Genera 8.0, you will see the following error:

```
Trap: The variable SI:*TIMER-QUEUE* is unbound.
While in the function SI:PRINT-TIMER-QUEUE ← (:INTERNAL
SI:COM-SHOW-TIMER-QUEUE 0) ←  CP::WITH-STANDARD-OUTPUT-BOUND-INTERNAL
```

unless some program has created an old-style timer.

Usually the right way to recover if you have reached this point is to press ABORT to return to command level.

Note that old-style timers are still supported for compatibility, but you should never use them in your new applications, and you should convert your old code to use new timers wherever possible and convenient. For more information, see the section "Timers".

Changes to Zwei and Zmacs in Genera 8.0

Changes to Zwei and Zmacs Since Genera 7.2

Improved User Interface for Changing Character Styles in Zmacs

Genera now includes a new, optional user-interface mechanism for specifying a new character style, for all the Zwei commands that change the character style of text, such as c-J, m-J, c-m-J, and c-X c-J.

The new mechanism is an option that is selected according to the value of a new Zwei variable, **zwei:*change-style-mode***, or Change Style Mode. The variable has two possible values, Quick and Prompt For Name. Prompt For Name is the default, and selects the existing interface for specifying character styles in the minibuffer. You do not activate the new behavior until you specify Quick mode.

Previously, you had to type a line of keyboard input to the minibuffer to specify a character style. The new alternative (Quick) mode allows you to define your own one-character abbreviations for commonly used styles, which can then be specified by typing just one character at the prompt. It comes with an initial set of abbreviations (I for italic, B for bold, and so on). It provides both an escape mechanism to be prompted for a full character style and a mechanism for redefining existing abbreviations and defining new ones on the fly.

You set the variable by a Zmacs command: m-X Set Variable / Change Style Mode / Quick. If you want to establish this setting routinely in your init file, use this form:

```
(ZWEI:SET-ZWEI-VARIABLE "Change Style Mode" :QUICK)
```

You can also make your own standard abbreviations in your init file. For example, the following establishes T as an abbreviation for NIL.NIL.TINY:

```
(ZWEI:SET-CHARACTER-STYLE-DISPATCH #\T '(NIL NIL :TINY))
```

Typing any of the style-changing commands prompts for one character. Possibilities are:

Character	*Description*
c-G, ABORT, RUBOUT	Aborts
ESCAPE	Prompts for a style (like the old Zwei behavior)
RETURN or SPACE	Accept the default (from most recent change)
char	First time, asks for a definition and uses it; after that, uses the existing definition
m-*char*	(Re)Defines the action of *char*

You use **zwei:set-character-style-dispatch** to set the abbreviation defaults. The following definitions are predefined:

Abbrev.	*Style*
B	NIL.BOLD.NIL
I	NIL.ITALIC.NIL
P	NIL.BOLD-ITALIC.NIL
N	NIL.NIL.NIL
S	NIL.NIL.SMALLER
L	NIL.NIL.LARGER

The major incompatibility between Prompt for Name and Quick is that pressing m-J RETURN in Quick mode does *not* produce NIL.NIL.NIL; you must press m-J N to do that.

Another difference is that Quick mode makes it possible to use style-change commands in the minibuffer. This is not possible in Prompt for Name mode, because entering the minibuffer recursively is not allowed. Note that you do not see the prompt if the minibuffer is exposed.

Miscellaneous Changes to Zwei and Zmacs Since Genera 7.2

The first four items below were Genera 7.2 ECOs.

- Zwei's redisplay no longer tries to display a character outside the left edge of the window.

- Switching buffers in Zmacs once again sets **zwei:*major-mode***.

- Point no longer alternates from top to bottom during m-V in Zwei.

- The Zmacs Undo/Redo commands now write-lock the interval being modified.

- The function **zwei:save-all-spell-dictionaries** now has an optional **ask** parameter. If **ask** is **t**, the function prompts for permission to save each dictionary.

- The following Zmacs commands are now documented: Tabify, Untabify, and Fill Differently.

  ° The Tabify and Untabify Zmacs commands now work on a region, if a region is marked. If not, the commands affect the whole buffer.

  ° The Fill Differently Zmacs command (m-sh-Q) can now be used with an argument to specify the fill increment. If the fill-width is 48, then m-sh-Q c-5 m-sh-Q would fill to 48, then to 53.

- If you abort a source-compare-merge with c-ABORT, the streams (which might be files) being compared are closed. In previous releases, the streams were left open.

- In previous releases, there was an inconsistency in how c-Q behaved when given in a normal Zwei window versus in the minibuffer. In normal Zwei, c-Q can be followed by up to three octal digits, which causes a character with that char-code to be inserted. c-Q now behaves that way in the minibuffer as well.

- The file-has-changed warning has been improved to be more informative when you refind a file and the newest version that is found is older than the one read in. This can happen when somebody deletes the previously latest version.

- A bug has been fixed in killing comments on lines of code. In the past, when you killed a comment on a line of code (c-m-; in Zwei Lisp mode), if the previous command was also a kill command, the killed comment was appended to the previous kill in the kill ring. This was incorrect behavior, and the result was a strangely scrambled text interval on the kill ring.

- A bug has been fixed in m-X List All Definitions and similar commands. Those commands produced mouse-sensitive output on the typeout window, and if you clicked on an item to edit the definition, the item following it was erroneously chosen to edit.

- The self-documentation of the String Search command (normally accessible via c-S END) in Zwei has been improved to mention the c-M subcommand, which grabs and appends marked text from the buffer to the search string.

- You are now warned and queried if you try to save a buffer that started as a non-file buffer, and you choose a pathname that already exists. This is probably an unrelated file that happens to have the same name.

- A bug has been fixed in c-sh-V (Describe Variable at Point). In the past, if you typed ***print-array*** followed by c-sh-V, the result stated that the value is **t**; if you then typed SPACE, the variable would be evaluated as **nil**. Describe Variable at Point now evaluates any variable before binding any specials for printing the value, because the variable might be one of those specials.

- Previously, the Dired F command refused to format any file that did not have a file type of .text or .mss. It now tries to format any 8-bit file type that does not have a binary-file-byte-size, or belong to a list of known hybrids.

- If you try to read a file in a nonexistent directory into a Zmacs buffer, you are now asked if you want to create the directory. In the past, an error was signalled, and you would have to do m-X Create Directory, then repeat the c-X c-F.

- The Zmacs command Down Comment Line (m-N) now takes account of ***comment-end*** when looking for the empty comment, since ***comment-end*** is not necessarily the empty string for all languages. This fixes a bug in languages such as C and Pascal, where m-N and m-P (Up Comment Line) did not delete the empty comment on the current line.

- Zwei character replacement handles character styles more consistently. For example, c-% and m-% now replace ")" with "]" in "**[sic)** semper" to yield "**[sic]** semper", rather than "**[sic**] semper".

- m-X Reap File and m-X Clean Directory no longer fail immediately when given very long file names.

- If you give a numeric argument to the c-m-X Edit Buffers command (to prompt for an apropos string), then specify a string that matches no buffer names, Zmacs no longer errs while reverting the *Edit-Buffers-N* buffer.

- The m-X Split Screen command now works properly when you have enough buffers (about 80) to make the buffer menu scroll, position the mouse near the vertical center of the screen, and click on a buffer choice.

- Very long buffer names now display correctly in the menu of buffers that is popped up for m-X Split Screen.

- The contents of a Dired buffer are no longer lost when you abort out of a Revert Buffer operation.

New Features in Zwei and Zmacs in Genera 8.0

New Key Bindings in Genera 8.0

- Reverse Lines is now available on m-X rather than c-m-X.

- m-X Kill or Save Buffers is now bound to c-X c-m-B. Giving a numerical argument to c-X c-m-B inhibits the initial marking of the menu that Kill or Save Buffers pops up.

- c-m-X Edit Buffers is now bound to c-X c-sh-B.

- Canonicalize Whitespace is now bound to c-X c-SPACE.

New Commands in Edit Buffers in Genera 8.0

Two new commands have been added to c-m-X Edit Buffers (c-X c-sh-B):

E Immediately selects the buffer at the point for editing, without performing any marked actions. (You can use c-m-L to return to Edit Buffers.)

= Compares the buffer to its corresponding file.

New Command: End of Code Line

There is a new command, End of Code Line, that is bound to c-RETURN. It positions the cursor at the end of the last "real" (non-comment) character on the current line. A numeric argument specifies by how many lines to move.

New Option for Add Patch Changed Definition Commands

m-X Add Patch Changed Definitions of Buffer and m-X Add Patch Changed Definitions of Tag Table now offer a fourth choice, Ignore (prompt: I) along with Yes, No, and Proceed. Ignore suppresses further queries about the changed definition until you change it again. This is intended to eliminate queries about definitions you have changed but do not want to patch.

Improvements to Zwei and Zmacs in Genera 8.0

Improvements to Source Compare in Genera 8.0

- Source Compare can now compare more things:

  m-X Source Compare now prompts for additional quantity types in addition to File (F) or Buffer (B). The full list of quantities accepted (and their corresponding characters) are:

B	Buffer
D	Definition

F	File
K or c-Y	Last Kill
P or m-Y	Previous Kill
R	Region

If R is used but no region is visible, you are prompted for a buffer name and then placed in a recursive edit on that buffer in order to establish a region.

Using R turns off the region, so using R twice in the same command will *always* provide the recursive edit to place the second region, even if a region was in place at the outset of the command. This makes it possible to compare one region to another.

In a recursive edit to establish a region, if you exit without establishing a region, an attempt is made to use the definition at the point instead.

- Source Compare now reports to you which *Source-Compare-*N** buffer it has used for the results of comparison. In most cases, the buffer name in this message is mouse-sensitive, so you can click Left on the name to select the buffer.

Miscellaneous Improvements to Zwei and Zmacs in Genera 8.0

- When you changed a buffer's default typein character style, the mode line previously displayed:

```
Style:  [FOO.BAR.BAZ]
```

  It now displays:

```
[FOO.BAR.BAZ]
```

- Reading Lisp files into the editor is faster. The parsing done by Lisp sectionization runs about 2.5 times faster, and completion **aarray** sorting is about 15-20 percent faster.

- Two long-standing problems with **zwei:read-directory**, which prompts for a directory pathname in such commands as m-X Dired, has been fixed. They both derived from the odd way that the function tests to find whether the default pathname is a directory pathname or a file pathname, in order to transform it for the offered default of the prompt. The function now traps all errors and uses **signal-proceed-case** to present an understandable error when there is some extraneous failure.

- m-X Show Character Styles now tells you about the font of non-printing characters that occur invisibly in the region.

- A new variable, **zwei:*revert-unedited-buffers-for-new-versions***, now controls the behavior of Refind File, Refind All Files, and Revert Buffer. Its default is **:query**. You can set it to **:always** to have the newest version of a file picked up without asking you about it. See the section "Zmacs Customization in Init Files".

Incompatible Changes to Zwei and Zmacs in Genera 8.0

Change to Default Arguments for Change Definitions Commands

m-X List Changed Definitions, m-X Edit Changed Definitions, m-X List Changed Definitions of Buffer, and m-X Edit Changed Definitions of Buffer now default their arguments in a different way. The new default is "since the file was last saved", rather than "since the file was last read". This is intended to make using the Add Patch commands easier.

Changes to Zmail in Genera 8.0

Changes to Zmail Since Genera 7.2

Zmail Filtering Improvements

- In the filter definition menu, search keys can use the extended search characters, "←" and/or "→", or can be interpreted as ordinary strings.

  In the filter selection menu, only ordinary strings are accepted.

  To use the extended search characters in the search key, you use c-H while entering the key; c-H HELP works. If you use extended characters, the entire address (including any domain name after the "@") is tested for a match.

  To use "←" and/or "→" in the search key, "←" must be the first character of the key and "→" must be the last character of the key. (If these characters appear anywhere else in the key, they are ignored). In this case, only the local part of the address is tested for a match (that is, the domain name after the "@" is ignored). If the key is "←TEXT→", an address matches if TEXT is found anywhere in the local part; if the key is "←TEXT", an address matches if the local part ends with TEXT; if the key is "TEXT→", an address matches if the local part begins with TEXT.

  If the search key is an ordinary string, it may contain "!"'s, "%"'s, and "@"'s.

  If the key contains an "@", the local part of the address is tested for an exact match against the left part of the key and the address's domain is tested for an exact match against the right part of the key; if the key doesn't contain an "@", only the local part of the address is tested.

If the key contains "!"'s or "%"'s, the local part of the address is tested for an exact match against the key.

If the key doesn't contain any of these special address characters, any use of "!" and "%" in the local part of the address is properly stripped before testing the remainder of the local part of the address against the key.

Examples:

```
"←Zmail" matches "Info-Zmail@Elephant-Butte"
         but not "Zmail-Tricks@Alderaan".

"Palter" matches "Palter@Alderaan",
                 "Palter%Alderaan@Riverside", and
                 "riverside!alderaan!palter@ARPA-Gateway.UUCP".

"Palter@Alderaan" matches "Palter@Alderaan"
                  but not "Palter%Alderaan@Riverside" or
                  "riverside!alderaan!palter@ARPA-Gateway.UUCP".
```

- In filter definitions, Zmail previously recognized "←TEXT→" as an address search key that matched addresses whose local part contained TEXT anywhere. In addition, "TEXT→" and "←TEXT" in the address search keys used in filter definitions now match addresses whose local part begins with TEXT or ends in TEXT, respectively.

- Previously, you could not use any form of filtering to select messages in "standard" digests format. (Said messages are known simply as digests.) *Digest* is now a filtering option in the filter selection and definition frames.

TENEX/TOPS-20 Mail Files Are More Robust

Zmail's reading and parsing of TENEX/TOPS-20 format mail files is now protected against improper byte counts and other malformed message status lines.

An improper byte count can cause Zmail to decide that the next message starts in the middle of a message; now, Zmail treats the entire contents of the partial message as a single message rather than treating each line (or two) as a separate message.

Also, if an improper status line is encountered, Zmail now permanently marks the message as unparseable and provides a reasonable explanation for the m-X Explain Bad Headers command.

New Zmail Commands

- m-X Start of Summary Window (c-m-<) and m-X End of Summary Window (c-m->) move the summary window to the beginning and end, respectively, of the current sequence without affecting the current message.

- Previously, if a sequence contained many digests, there was no simple mechanism to run the Undigestify command on them; each digest had to be dealt with manually.

  A new command, m-X Undigestify All Messages (also available as [Undigestify] on the [Map Over] menu), is equivalent to using the Undigestify command on each message in the current sequence that is a digest.

Miscellaneous Changes to Zmail Since Genera 7.2

- A Genera 7.2 ECO made the m-X Edit Mail File (c-X c-F and [Select(R) / Read/Create File]) and m-X Examine Mail File (c-X c-R and [Select(R) / Examine File]) commands use the current message's first file reference, if any, as the default when asking for the file to be loaded into Zmail. This change has been reverted in Genera 8.0, so the default for each command is as it was originally in Genera 7.2.

- Genera 8.0 provides support for software ECOs.

  - A new command, m-X Decode ECO, has been added.

  - Support has been added for binary-to-"ASCII" file encoding and decoding, which can be used to distribute future ECOs via mail.

  These were Genera 7.2 ECOs.

- In earlier releases, the Hardcopy All ([Map Over(R) / Hardcopy]) command asked the "too many messages" question before asking for the hardcopy options and did not distinguish between actually hardcopying the messages or just hardcopying the message summaries when asking the question.

- When hardcopying message summaries in landscape mode, Zmail truncated the summary lines as if it were producing portrait output.

- Previously, pressing c-m-0 c-m-V too quickly after issuing a Get New Mail from Inbox (G) command could fail. This bug has been fixed.

- Zmail's address parser now accepts "User@DOMAIN-1@DOMAIN-2" in addition to "User@HOST-1@HOST-2" even though, technically, multiple "@"'s in addresses are no longer valid.

- If you click on a message in the summary window after clicking on [Author], [Author/Recipient], or [Recipient] in the filter selection menu, and there is only one possible address, a small confirming menu now pops up.

- Clicking m-Left on a file reference in the Zmail header window now edits the file in an available Zmacs.

- In Genera 7.2, if you clicked on [Edit] in the profile editor to edit the profile directly and then used incremental search (c-S or c-R), the cursor in the profile buffer could disappear while the incremental search is in progress. This bug has been fixed.

- In earlier releases, when Zmail was given an expiration date, it did not use the default value as the base time for parsing. As a result, if you entered an incomplete date (that is, no year), it was interpreted relative to today, which could cause the expiration date put on the message to be earlier than the message's start date. (For example, if today were "11 March 90" and the start date were "12 April 90", the default expiration date would be "13 April 90". If you entered "the 14th", Zmail used "14 March 90" as the expiration date rather than "14 April 90".) This bug has been fixed.

Improvements to Zmail in Genera 8.0

- Zmail can now read and write mail files in the Macintosh filesystem.

- The Zmail file default for UNIX hosts is now a file named "RMAIL" that is parsed as a Babyl file for compatibility with GNU EMACS. This means that the same mail file can be read with both GNU EMACS and Zmail.

- Occasionally a message's header information does not properly reflect the character styles used in the message. When you try to save this message, Zmail would believe it was sending a draft message and signal a non-restartable error. Zmail now correctly signals an error which has a proceed option that lets you fix the message and retry saving the mail file.

- Occasionally you see a message reference in a message header like this:

  ```
  Yankee Doodle's message of 04-Jul-76 <17760704.1@U.S.A>
  ```

  In earlier releases, such messages were not parsed usefully. As a result, the messages containing these references would not get linked into the correct conversation, and conversations would become more fragmented than they might be otherwise.

  These message references are now parsed correctly.

All mail file types except KBIN will automatically see the effect of this change when loaded into a world containing this fix because messages are automatically reparsed every time the mail file is loaded. However, in order to avoid a KBIN version incompatibility over so small a fix, KBIN files will not be automatically updated. If you have old messages in KBIN files which you would like to be reparsed, you can load those KBIN files into Zmail and use the command c-U m-X Reparse All Loaded Messages to reparse the messages in all loaded KBIN files.

- The L Zmail command ("Label" Message) that prompts in the minibuffer for a keyword to add to the current message now maintains a history, so you can use c-m-Y and m-Y to yank back keywords previously typed in that minibuffer.

Changes to Files and Streams in Genera 8.0

Changes to Files and Streams Since Genera 7.2

New Feature: UNIX Printers From Genera

You can now use UNIX-spooled printers from Genera. See the section "Installing and Using UNIX Printers From Genera".

New Canonical File Type: IBIN

Genera now includes a new canonical file type, :IBIN, for files compiled for Ivory-based systems. :IBIN has been added to the variable **fs:*known-types***.

The canonical type for Ivory-compiled files is :IBIN. The extension is normally .IBIN, but on a system type of :UNIX, the extension is IB, and on :VMS and :MSDOS it is IBN.

Attempting to load an Ivory-compiled file on a 3600-family machine or vice versa results in an error.

Data (not containing compiled code) dumped with **sys:dump-forms-to-file** or **compile-file** can be read/written interchangeably by Ivories and 3600s.

See the variable **si:*default-binary-file-type***.

Changes to the Serial I/O Facility

Serial is now a separate system.

Changes to the Serial I/O Facility for Genera 7.4 Ivory

The Serial I/O Facility was rewritten for Genera 7.4 Ivory. It now works as documented.

You should note the following minor changes:

- The **uss** package has been removed. Users should not have been using symbols in this package. However, if you are using any of these symbols, such as **uss::*serial-interfaces*** and **uss::serial-port-lock-holder**, you should change your code to use the documented interfaces.

- It used to be possible to create your own flavor of serial stream and supply all the options to **si:make-serial-stream** in the **:default-init-plist** option to **defflavor**. Now you can supply stream-specific options only to **:default-init-plist**; others must be supplied with **:after** methods. See the section "Creating Your Own Flavor of Serial Stream".

- It should be noted that the serial chip used by the XL400 has a slightly different behavior with respect to parity checking than the chips used in earlier Symbolics machines. The 68562 chip checks parity *before* checking for XON/XOFF characters, the older chip checked *after*. This means that you have to be more careful about parity settings than previously. If the parity is wrong, XON/XOFF flow-control does not work.

Designation of Serial I/O Ports for the UX-Family

The UX-family uses the serial I/O ports of the SUN that is its embedding host. These ports are designated by their device name in UNIX, for example, `"/dev/ttyA"`. The **:unit** parameter for serial streams now accepts this naming scheme as well as the integer naming scheme for the other hardware platforms.

New Facility for Data Compression

It is now possible to compress data when writing it to some storage medium and to decompress it upon reading.

Currently, only a compression and decompression substrate exists. No user-level interface is supplied, though that will change in a future release. The only current users of the compression substrate in 8.0 are the Distribute Systems and Restore Distribution frames; compression is used in these frames to decrease the amount of tape required to make a distribution. In a future release, user-level commands (such as CP commands) to compress and decompress files will be provided, as will be the ability to read a compressed file and automatically decompress it upon reading the way that styled characters are currently converted.

Note: The Compression Substrate is preliminary in Genera 8.0. It might be radically altered in function and/or interface in future releases. For information about this new facility, see the section "Compressing Data - the Compression Substrate".

Symbolics is interested in knowing what uses customers make of the compression substrate, and where additional functionality is desired. Note that, since only a compression substrate, and not a user interface, is provided in 8.0, you should clearly differentiate between a request for a user interface (which will most likely be provided in a subsequent release) and augmentations to the substrate.

Miscellaneous Changes to Files and Streams Since Genera 7.2

- The **:directory-pathname-as-file** method for TENEX now returns the correct version number. This was a Genera 7.2 ECO.

- Restoring a LMFS file for which both an original and a renamed version already existed on disk caused an infinite loop. This bug has been fixed. This was a Genera 7.2 ECO.

Improvements to Files and Streams in Genera 8.0

Improvement in Performance of Character I/O Streams

Character I/O streams are three to eight times faster in Genera 8.0.

LMFS Reloader Now Works with Remote Hosts

The LMFS reloader can now be used to reload onto the LMFS of a host other than the one where you are using the Reloader menu.

You specify a remote host as the target by preceding the pathname with *remote-host-name:* .

See the section "Reloading and Retrieving LMFS Files".

Serial I/O Implementations on MacIvory and Symbolics UX400S

- MacIvory serial I/O has been reimplemented for Genera 8.0. It is now more reliable and can drive devices at full speed without tying up the Macintosh.

- You can implement serial communications on your Symbolics UX-family machine in Genera 8.0. The unit numbers on the UX-family machine are the names of the UNIX serial ports. Note that you must determine the serial communication parameters for the UNIX part of your UX machine before attempting serial communication. The **si:make-serial-stream** function does not perform any validity checks.

Incompatible Changes to Files and Streams in Genera 8.0

Changes to with-open-stream-case and Related Macros

The macros **with-open-stream-case**, **with-open-stream-case-if**, **with-open-file-case**, and **with-open-file-case-if** have been changed to allow the **:no-error** clause, if specified, to return multiple-values.

In previous releases, the :**no-error** clause could only return multiple-values when a **return-from** or **return** form was used to return values from the **:no-error** clause.

This change may affect your existing code and could introduce new behavior. Check your calls to any of the above macros where a **:no-error** clause is specified. If the **:no-error** clause returns multiple values that will now be passed back to the caller, make sure the caller handles the multiple values correctly.

Known Problems with Files and Streams in Genera 8.0

Synchronous serial does not work in Genera 8.0. Contact Symbolics Software Support for a patch to fix it.

Changes to User Interface Programming Facilities in Genera 8.0

Changes to the User Interface Programming Facilities Since Genera 7.2

New Keyword to tv:with-output-to-bitmap and tv:with-output-to-bitmap-screen

The functions **tv:with-output-to-bitmap** and **tv:with-output-to-bitmap-screen** now accept the **:host-allowed** keyword.

:host-allowed Allows the use of host storage and drawing primitives for an off-screen window stream. The default is **nil**.

Note that the **tv:with-output-to-bitmap** version is less useful, since in order to get its value the system will need to copy the final result back into an actual guest bit array. Within the **tv:with-output-to-bitmap-stream** form, the **:bitblt** messages can be used to copy to (and from) the screen entirely on the host side.

New Function: tv:add-dispatching-select-key

This function makes it possible to specify a single select key for a program, and then choose the program based on run-time conditions. This dispatching feature enables you to have more than one program associated with a select key.

See the function **tv:add-dispatching-select-key**. (Note that in Genera 8.0 this function is documented online, but not in the printed documentation set.)

New User Interface Programming Functions

tv:%screen-allocate-sheet-temporary-bit-array *basic-screen sheet width height*
This method on screens allocates a bit array suitable for use with **tv:sheet**.

tv:%screen-deallocate-sheet-temporary-bit-array *basic-screen sheet pixmap*
This method on screens deallocates the array returned by **tv:%screen-allocate-sheet-temporary-bit-array**.

tv:with-temporary-sheet-bit-raster *(raster-var sheet width height)* &body *body*
Binds *raster-var* to an array suitable for use with sheet of size width times height.

tv:with-off-screen-drawing *(sheet* &key *:complete-redisplay)* &body *body*
Copies the contents of *sheet* into off-screen memory, executes *body*, and then copies it back onto the screen.

Miscellaneous Changes

- **tv:prepare-sheet** now uses locks instead of using **without-interrupts**, so multiprocessing can now occur during window drawing. This will not require you to make changes to your programs, but might prove helpful for you to know during debugging.

  The change was made to enable talking to remote windows of all kinds (Macintosh windows in MacIvory, X Windows, and so on) where the medium for communicating with the window inherently involves multiprocessing so that **without-interrupts** will not work.

- The **type-or-string** presentation type now takes a *:reject-null-string* keyword argument. A non-null value allows you to give no default value, but still not allow a null (string) as input. A null value means that this presentation type refuses a null string only if there is a non-null default. This keyword enables you to control whether or not a null string is allowed as input separately from the default value.

- The following were Genera 7.2 ECOs:

  ° Adding or removing presentations from the middle of a dynamic window (including a Document Examiner window) took too long to move the so-called "boxes" around. This task has now been sped up by an order of magnitude or more.

  ° Updating frame options and parsing new **define-program-framework**s could fail, as on **:extend-width**. This has been fixed.

- **dw::ptypep** now works correctly for types in which some subtype appears more than once, such as **'(or list list)**.

- **:output-as-presentation-1** sets the proper superior on simulated presentation recording streams, to support some new Concordia features.

- Dynamic Windows mouse-handling has been improved:

  - Typeahead on an unselected window that would otherwise cause infinite loops is now ignored.

  - Some timing windows that could cause highlighting boxes to stay highlighted too long have been closed.

  - Another timing window has been closed that could cause click-ahead to use the wrong modifier keys.

  - Anonymous graphical presentations are now handled correctly.

- After the output history has been cleared asynchronously, the Command Processor prompt is now always refreshed.

- Graphical output to an empty window is now done correctly.

- Left kerns in **graphics:draw-string-image** are now handled properly.

- Redrawing of some Accept Variable Values (AVV) menus has been sped up.

- **tv:with-output-to-bitmap** now handles text correctly.

- **tv:with-this-mouse-grabbed-internal** now handles the second mouse documentation line correctly.

- Random notifications no longer lock up the window system while Change Variable Values menus are being used.

- You can now refresh a non-dynamic pane in a program frame without getting an error.

- Frame-Up now inserts forms into the editor using a standard printing environment.

New Features in the User Interface Programming Facilities in Genera 8.0

- You can now allow a program to change the name of the `(no window)` state by setting a variable, **tv:*no-window-alternate-wholine-string***.

- **formatting-graph-node** has a new option, **:drawing-mode**. For more information, see the section "**:drawing-mode** Option to **formatting-graph-node**".

Incompatible Changes to the User Interface Programming Facilities in Genera 8.0

Previously, **tv:add-to-system-menu-windows-column** was misspelled with an "s" after "system". The misspelling has been corrected in this release; if you call the function by the old, incorrect name, you will get a warning.

Changes to Site Operations in Genera 8.0

Changes to Site Operations Since Genera 7.2

Support Added for Hayes Modems

Genera now includes Dialnet support for Hayes modems.

See the section "Using a Hayes-1200 Modem with Dialnet" and see the section "Using a Hayes-2400 Modem with Dialnet".

Changes to the Mailer

- The following Mailer bugs have been fixed.

  - When receiving incoming mail, the Mailer could leave message objects allocated forever if the sending system aborted the transmission at almost any point.

  - When cold booting, the Mailer sometimes left old message objects allocated forever.

  - When warm booting, the Mailer lost the messages that were actively being transmitted at the time of the Mailer crash. (The messages reappeared at the next cold boot.)

- The Mailer no longer requires valid copies of the files `Mailboxes.text` and `Forwarding.text` when cold booting. After each successful reading and "compilation" of the mailboxes table, the Mailer now writes an image of the table into a snapshot file, `>Mail>Dynamic>Mailboxes.Snapshot.` At cold boot time, the Mailer now tries to read Mailboxes.Snapshot first. If, for any reason, this fails, the Mailer reverts to the old behavior of reading and "compiling" `Mailboxes.text` and, if present, `Forwarding.text.`

Since the binary image is always written after successful "compilations" of the mailboxes table, it always reflects the latest working state of the mailboxes table and, therefore, is always the proper file to use when cold booting. If `Mailboxes.text` and/or `Forwarding.text` is changed during the interval between the last "compilation" and a cold boot, the Mailer notices this fact immediately once in operation and performs its normal background update of the mailboxes table.

This was a Genera 7.2 ECO.

- Two new options, settable in `Options.lisp`, control the maximum size of Mailer log files and the number of log files that are retained:

  ° **mailer:*log-file-maximum-size*** (default is 450K characters)

  ° **mailer:*log-file-generation-retention-count*** (default is 10)

  A new Mailer log monitor program command, Switch Log Files, starts a new log file.

- The Mailer frame's Queues pane now displays the status of two pseudo-queues — Active and Retransmit — in addition to the three queues (Incoming, Outgoing, and Slow) that it displayed previously. The active pseudo-queue contains those messages that are actually being transmitted by one of the Mailer processes. The retransmit pseudo-queue contains those messages that are awaiting retransmission and are not in any of the other queues.

- The Mailer's Resolve Hosts command now also accepts either a sequence of hosts or the token "All". Previously, this command did not accept arguments and always tried to resolve all unresolved hosts.

- A new command, Probe Hosts, has replaced the Mailer's Probe Host command. This new command accepts either a sequence of hosts to probe or the token "All", which causes it to probe all hosts with queued mail. The documentation on the Mailer has been improved. It now describes the Mailer Operations window, how to perform administrative tasks related to the Mailer, and the contents of the configuration files used by the Mailer. The installation instructions have also been improved.

  For general information on the Mailer, see the section "Symbolics Store-and-Forward Mailer".

  For installation instructions, see the section "Installing and Configuring the Mailer".

Miscellaneous Changes

Previously, there was no check for whether an object being deleted from the namespace was referenced by other objects. If an object with references to it was deleted, it caused errors on the namespace servers later on. Now, a primary name server checks for references to the object that is to be deleted, signalling an error if references exist and reporting which objects have references.

For more information, see the section "Delete Namespace Object Command".

After the Set Site operation, there is no longer a notification of an error in an NFS background operation.

Improvements to Site Operations in Genera 8.0

New Set Site and Define Site Dialogues

The Set Site and Define Site CP commands have been completely rewritten to use the user interface paradigm that is consistent with the rest of the system and to provide an expanded interface for a wider variety of system configurations.

For more information:

> See the section "Set Site Command".
> See the section "Set Site Dialogue".
> See the section "Define Site Command".
> See the section "Define Site Dialogue".

si:fix-fep-file Now Works on Ivory-based Machines

si:fix-fep-file works on Ivory-based machines in Genera 8.0.

Changes to the Copy World Command in Genera 8.0

The Copy World Command offers new proceed options on the RESUME key in cases of network problems that stop a copy in the middle, depending on how far the copy operation has proceeded:

- Copy the rest of the band and checksum the entire band.

- Copy the rest of the band.

- Checksum the rest of the band.

In each case a proceed option to retry the whole operation is also offered.

Adding Multiple Services to Hosts

It is now possible to add more than one service at a time to a host or set of hosts. The commands Add Service To Hosts and Remove Service From Hosts have been renamed to Add Services To Hosts and Remove Services From Hosts.

Users Are No Longer Prompted During Booting

No prompting of the user should be required during booting. If a condition exists that previously required the user to be asked a question, the system will fall back to a known state and a notification will be issued.

time:initialize-timebase has been changed so that the user is not prompted to enter the current time unless the calendar clock on the machine has not been initialized or no network time server responded. During booting and network reset, the current time is obtained in the following manner:

- For embedded systems, the current time is obtained from the embedding host. For example, on a MacIvory, the Macintosh is queried for the time of day, and if a valid time is obtained, the Ivory uses it.

- For standalone sites that do not consist of an embedded system, the hardware calendar clock is used to obtain the current time.

- Otherwise, the machine sends out a broadcast on the primary network for the current time. If no host responds, the calendar clock is used. Only if the calendar clock is uninitialized is the user prompted to enter the current time.

The Set Time Command Processor command can be used to change the calendar clock after the machine is booted.

In previous releases, when the local host was unable to verify its identity via a broadcast, the user was presented with a prompt similar to: "Is the local host still Foo at ACME-CORPORATION?". This prompt has been eliminated. If the local host cannot verify its identity, it uses the identity cached in the world, if there is one. Otherwise, the host reverts to the distribution site so the user can use Set Site to set the proper site. This change should be of no consequence at most sites, and will avoid the possibility of a host booting with a name such as "Chaos|25252" in an unknown site.

New Features in Site Operations in Genera 8.0

Namespace Cleanup Tool

In Genera 8.0, obsolete fields have been removed from some classes of namespace objects. These fields were declared obsolete in Genera 7.0.

There is a tool to go through namespaces and clean up objects with obsolete fields. The tool is documented in the file SYS:NETWORK;NAMESPACE-FIXES.TEXT. To use the tool, load the file SYS:NETWORK;NAMESPACE-FIXES.

Using the Internet as the Primary Network

Genera 8.0 allows machines with IP-TCP in their worlds to be booted using the Internet network as their primary network. The FEP command Set Network Address (Set Network-Address on 3600-family machines) or Set Boot Options :Network Address on Ivory-based machines is used to set the address in the FEP.

(Note: The Set Network-Address 3600-family FEP command is in the `-rel7` flod file, which must be scanned in the FEP before you can use the command.)

See the section "Set Network Address FEP Command".

Multiple addresses can be specified by separating the Network|Address pairs by a comma (for example, `Chaos|24407,Internet|128.81.41.7`).

If the namespace server is located on a different subnet than the local machine and the two are separated by an IP gateway, the address of the gateway must be specified after the Internet address. The syntax is specified in the section "Set Network Address FEP Command". This information is used to construct a temporary route until the proper information for routing can be obtained from the namespace server.

The subnet mask for the local network is assumed to be 255.255.255.0. If a different value is needed, it can be appended to the gateway address by a semicolon. The syntax is specified in the section "Set Network Address FEP Command".

Genera 8.0 fully supports access to namespace information via Internet protocols. The Namespace protocol is implemented over the TCP medium using port 261 and the Namespace-Timestamp protocol is implemented over the UDP medium using port 261.

Note: Please do not use Internet-based namespace services unless all the Symbolics machines at your site are running Genera 8.0. Machines running Genera 7.4 Ivory and earlier releases will try to use the TCP and UDP (instead of Chaos) protocols for the namespace services and will encounter difficulties during booting.

New Namespace Editor Mode

There is a new mode in the namespace editor, Insert Defaults. You can enable this mode with the Namespace Editor command Insert Defaults or by keywords to the Namespace Editor commands Edit Object, Create Object, and Copy Object, or with the Command Processor commands Edit Namespace Object and Create Namespace Object. When Insert Defaults mode is in effect, default values for some indicators are inserted.

For more information, see the section "The Insert Defaults Namespace Editor Command".

New Service: `TAPE TCP RTAPE`

TCP Rtape has worked since Genera 7.2, but is not documented in the tape or TCP documentation. Rtape over TCP is more reliable than Chaos Rtape, and should be the tape service of choice for any site that runs IP-TCP.

New Variable: su::*log-monitor-window-lines-to-keep*

su::*log-monitor-window-lines-to-keep*controls the behavior of logs written by facilities such as the Mailer and Domain Name Server. For more information, see the variable **su::*log-monitor-window-lines-to-keep***.

Incompatible Changes to Site Operations in Genera 8.0

Dialnet Registries Are Obsolete in Genera 8.0

- Dialnet Registries are now obsolete. You should convert your Dialnet Registry to the Domain Name System. Instructions for this conversion are provided in *Site Operations*. See the section "Converting From Dialnet Registries to the Domain Name System".

- The Mailer pane's Reload Dialnet Registries command has been replaced by a new command, Reload Subnet File.

- The Dialnet documentation includes new examples that show how to set up an integrated configuration of the Mailer, the Domain Name System, and Dialnet. See the section "Writing Dialnet Configuration Files".

Avoiding a Race Condition Between the Domain Name System and the Mailer

The Mailer depends on the Domain Name system to operate properly, particularly for Dialnet hosts. However, when enabling services, there can be a race condition between the Domain Name System being completely enabled and the mailer starting to process mail.

To avoid this, you may wish to insert the following code (which should be in the **mailer** package) at the end of your `>Mail>Static>options.lisp` file. This code causes the Mailer to wait when it starts up until the Domain Name system is completely initialized. **Note:** You should do this only if you run the Domain Name server on the same host as the Mailer. If you have many entries in your domain name files, you may wish to increase the timeout appropriately.

```
;;; Wait this many seconds for the DNS to come up.
;;; If NIL, don't wait at all.
(defconst *SECONDS-TO-WAIT-FOR-DNS-STARTUP* 30)
```

```
(defun SYNC-WITH-DOMAIN-SERVER ()
  (when *seconds-to-wait-for-dns-startup*
    ;; If you set NOTIFY-HOSTS, you should include the next line so
    ;; the hosts are parsed before this function runs (otherwise,
    ;; they're parsed afterwards, when this file finishes loading).
    (mailer:parse-notify-hosts)
    ;; Wait for the domain name server to start up.
    (process-wait-with-timeout "Await Domain Server"
                               (* 60 *seconds-to-wait-for-dns-startup*)
      #'(lambda ()
          neti:*domain-server-enabled*))
    ;; Tell somebody if the timeout was too small.
    (unless neti:*domain-server-enabled*
      (su::log-event
        :problem
        "Domain name server didn't start after ~
         ~A second~:P; starting mailer anyway."
        *seconds-to-wait-for-dns-startup*))))

(sync-with-domain-server)
```

Changes to the FEP in Genera 8.0

Changes to the FEP Since Genera 7.2

Changes to the FEP since Genera 7.2 all involve the Ivory-based, software-resident FEP, which supports all the functionality of the 3600-based, EPROM-resident FEP, and includes some additional functionality. The following information, with one exception, applies only to the Ivory FEP. The exception is the information on boot files, which applies to both the Ivory and 3600 FEPs.

Changes in the Contents of Boot Files

There has been a change in recommendations regarding which FEP commands should be given in the `hello.boot` file and which should be given in the `boot.boot` file on both Ivory-based and 3600-family machines. For more information, see the section "Contents of Hello.Boot Files and Boot.Boot Files".

Syntax Changes in Ivory FEP Commands

The Ivory-based FEP has two incompatible syntax changes in the way that FEP commands are invoked.

Ivory-based FEP commands do not have hyphens in their names. Hyphens are replaced by spaces. For example, the FEP command called "Declare Paging-Files" on a 3600-family machine is called "Declare Paging Files" on an Ivory-based machine.

Note: The syntax of FEP commands that take sequence arguments is different, depending on whether the machine is a 3600-family machine or an Ivory-based machine. A "sequence argument" is a list of things, such as the *sequence-of-filenames* argument to the Declare Paging Files and the Declare More Paging Files commands.

On 3600-family machines, the elements of the sequence must be separated with spaces. For example:

```
Declare Paging-Files file1 file2 file3
```

On Ivory-based machines, the elements of the sequence must be separated with commas and no spaces. (This syntax is compatible with the Command Processor syntax.) For example:

```
Declare Paging Files file1,file2,file3
```

New Features in the Ivory FEP Debugger

- Lisp's state is not lost by resetting/rebooting the IFEP (Ivory FEP). If the IFEP Debugger crashes while examining Lisp, you can restart the IFEP and try other IFEP Debugger commands.

- Most IFEP Debugger commands are similar to the commands of the regular Debugger.

- The IFEP Debugger includes the `c-M` command for capturing a backtrace which can then be accessed via the editor command `m-X` Insert Crash Data the next time you boot Lisp. (Note that this backtrace is lost if you reset the IFEP; send the backtrace to a FEP file first.)

- You can examine values of Lisp symbols in the IFEP Debugger.

- The IFEP Debugger includes DDT-like command accelerators that let you examine and modify memory locations. Use these commands with extreme caution.

New Commands in the Ivory FEP

Set Network Address FEP Command

Sets the primary network type and address of the local machine. On 3600-family machines, the syntax is Set Network-Address.

Set Boot Options FEP Command
: Sets default values for *keywords* on the local Ivory-based machine.

Find All World Files FEP Command
: Examines world files in all FEP directories, and updates the internal world database to include the worlds that it finds.

Create Initial FEP Filesystem FEP Command
: Builds an empty filesystem on a disk for use by Lisp (and the FEP).

Note that you should use the Create Initial FEP Filesystem FEP command with extreme caution.

New Features in the FEP in Genera 8.0

New Keyword to the Set Boot Options FEP Command

The Set Boot Options FEP command has a new keyword, :Autoboot on Halt.

Set Boot Options FEP Command

Note: This command is implemented only on Symbolics Ivory-based machines.

Set Boot Options *keywords*

Sets default values for *keywords* on the local Ivory-based machine. Use this command instead of entering the corresponding FEP command for each *keyword* in a `hello.boot` file.

The Set Boot Options FEP command makes it possible for users to boot distribution worlds without having to site-configure them first.

*keyword*s
: :Network Address, :Ethernet Address, :LMFS FSPT Unit, :Timezone Offset, :Timezone Name, :Site Name, :Namespace Descriptor File, :Default World, :Default Boot File, :IDS, :Autoboot on Halt, :Slave-Buffer Base

:Autoboot on Halt
: {Yes, No}. If Yes, and Lisp halts to the FEP, the FEP searches for an `autoboot.boot` file and processes it (if found) as if the system had just been powered up. If the autoboot process is aborted (by typing any character before it is finished) this option is set back to No (on the assumption that the system is no longer running unattended).

: Recommended usage: Your autoboot.boot file might look like this:

```
Hello
Set Boot Options :Autoboot on Halt Yes
(Type any character to abort autobooting...)
Autoboot Delay 20
Boot
```

Note that the Set Boot Options command must follow the Hello command, since it is in an overlay and would otherwise be unrecognized. (A "safer" `.boot` file would also specify the arguments to Hello and Boot.)

If you abort an autoboot, and later want to leave the machine unattended, you must reset the :Autoboot on Halt keyword. You could have it always set on in your standard hello file or boot file. Another way to do this is simply to halt to the FEP and reboot by:

```
Hello autoboot.boot
```

when you are ready to return to unattended operation. (Note that the default for Hello is always hello.boot, so this is safe to use with the above autoboot file where the Hello command reads its default. If you were to use Boot, instead of Hello above, you would end up recursively reading the autoboot.boot file. For safety, you should always fully specify arguments to commands in `.boot` files.)

:Default Boot File
Pathname for a boot file to use (must be in the local machine's FEPFS).

:Default World A world to load (must be in the local machine's FEPFS).

:Ethernet Address
The Ethernet address of the local machine. (This is used for DNA only.)

:IDS {Enable, Disable, Default}. How to set IDS when booting. The default is Default, that is, based on the world.

:LMFS FSPT Unit
The disk unit on which the File System Partition Table resides.

:Namespace Descriptor File
Pathname for a locally accessible namespace descriptor file (can be resident in the FEPFS, LMFS, or host (for example, Macintosh) file systems). If a site name is provided and it is not the current site, the site will be changed. If no descriptor file is provided, the site information is obtained from the network.

:Network Address
The primary network address of the local machine. A Chaos

address is specified as follows: `CHAOS|24623`. An Internet address is specified as follows: `INTERNET|128.81.41.147`.

:Timezone Name Name of the local machine's timezone.

:Timezone Offset Offset from GMT of the local machine's timezone.

:Site Name Site name for a site. Used to change sites at boot time.

:Slave-Buffer Base
Enables you to specify the base-address when it is something other than the default.

Load the Set Boot Options FEP command by scanning the overlay (flod) file `*-lisp.flod`. This command commonly appears in files of the type `hello.boot`.

For related information, see the section "Set Ethernet Address FEP Command", and see the section "Set Chaos Address FEP Command".

New Canonical Type for Boot Files

Previously, if you edited a .boot file and your default mode was set to Lisp, Zwei would issue warnings, as boot file buffers have no mode line. Now there is a new canonical type for `.boot` files, `:FEP-COMMAND`, defined to be like `TEXT`.

New FEP Command: Set Network-Address

There is a new FEP command for 3600-family machines, Set Network-Address. This command supports the new ability to use the Internet as the primary network. For more information, see the section "Using the Internet as the Primary Network" and see the section "Set Network Address FEP Command".

Change to Set Network Address FEP Command

The Set Network Address command now takes a comma-separated list as an argument.

For example:

```
Set Network Address CHAOS|12345,INTERNET|123.45.67.891
```

Contents of Hello.Boot Files and Boot.Boot Files

This section briefly describes the philosophy of which FEP commands belong in the `Hello.boot` file and which belong in the `Boot.boot` file on both 3600-family and Ivory-based machines.

Hello.boot File

This file should contain anything that holds constant for this machine, regardless of the world you are running. For example, this includes your flod files, your paging files, your network address, and so on.

First, this file must scan all the flods from which you want to use commands. This includes the lisp, loaders, info, and debug flods.

Next, the file should include the Initialize Hardware Tables command, so that any errors it encounters will be seen early in the booting process. (This command is completely optional in Ivory-based machines. It is automatically invoked by the Ivory FEP when needed, but errors it signals may be less clear when automatically invoked.)

Then, the `Hello.boot` file should declare any paging files you always want to use; this saves you having to manually Add Paging Files, or to say the same in your boot file. Do not declare any "spare" paging you want to save for an emergency, because Load World automatically uses all declared paging files, and thus the spare paging file would not be available in case of an emergency. It is harmless to declare paging files that do not exist (for example, one you might delete every now and then to make room for Copy World); nonexistent ones give a warning, but are not used.

Ivory machines only:

Starting with Genera 8.1, the `hello.boot` file for an Ivory machine should contain two lines:

```
Hello Innn
Hello Local (or hostname)
```

Each of these corresponds to a boot file. The *nnn* refers to the IFEP number, which is 328 for Genera 8.3.

Hello I*nnn* Boot File

The I*nnn*.boot file (where *nnn* is the IFep version number, which is 328 for Genera 8.3) should contain the commands to scan the flod files and initialize things.

```
Scan I328-lisp.flod
Scan I328-loaders.flod
Scan I328-info.flod
Scan I328-debug.flod
Initialize Hardware Tables
```

Hello Local Boot File

The `local.boot` file should contain those commands that set up this specific machine, declaring paging files, setting the network address, and any other boot options.

```
Declare Paging Files FEP0:>Paging-1.page
Declare More Paging Files FEP0:>Paging-2.page,Paging-3.page
Set Boot Options :Network Address Chaos|52525 :IDS Enable
```

Boot options are required for standalone sites. They are optional for other sites, but they do save manually invoking them or having them duplicated or inconsistent in boot files (if you have multiple boot files). See the section "Set Boot Options FEP Command".

The boot options replace other FEP commands. For compatibility, some of the boot options have corresponding commands. Note that some of these commands (such as Enable/Disable IDS on 3600-family machines), if present, must appear in the `Boot.boot` file, since they must follow Load World. (In the Ivory-based FEP, these commands warn if you try to use them at the wrong time, while in the 3600-family FEP they silently perform no operation if used at the wrong time). We recommend using the Set Boot Options command instead of the corresponding commands, so you can set all the boot options in one spot.

Normally, you set your network address with Set Boot Options, but you could use the Set Network Address command instead. Whichever way you choose to set the network address, it should be done in the `Local.boot` file, and should not be duplicated in your `Boot.boot` files.

3600 and Ivory Machines

Note that, in previous releases, you had to set the network address after loading the world, but this restriction no longer holds on either architecture.

Here is a sample `Hello.boot` file for a 3600-family machine:

Sample 3600-Family Machine Hello.Boot File

```
Scan v127-info.flod
Scan V127-loaders.flod
Scan v127-lisp.flod
Scan v127-debug.flod
Initialize Hardware Tables
Declare Paging-Files FEP0:>Paging-1.page
Declare More Paging-Files FEP0:>Paging-2.page,Paging-3.page
Set Chaos-Address 52525
```

(For a sample `Hello.boot` file for an Ivory-based machine, see the section "Sample Ivory-Based Machine Hello.Boot File".)

Boot.boot Files

These files should contain any setting that pertains only to the particular world (and microcode, on a 3600-family machine) this file loads and starts. (Note that this excludes the address of the machine.) There is a slim possibility that you might want to have some explicit paging file commands here (for example, a boot file to boot with just one paging file explicitly added for when you want to do world copying).

Sample 3600-Family Boot.Boot Files

Here is the sequence of commands for a Symbolics 3600-family machine that is cold booting a world from the local disk:

```
Clear Machine
Load Microcode FEP0:>3640-fpa-mic.mic.430
Load World FEP0:>genera-8-3.load
Enable IDS
Start
```

Here is the sequence of commands for a Symbolics 3600-family machine that is cold booting a world from a remote disk (netbooting):

```
Clear Machine
Load Microcode FEP0:>3640-fpa-mic.mic.430
Netboot inc-site-genera-8-3
Enable IDS
Start
```

The information in the rest of this section applies to Ivory-based machines only.

`Boot.boot` files for Ivory-based machines should contain only the following:

```
Load World
Start
```

Note that you can set your default world to boot in boot options. If you choose to load the default world, there is no need to give an explicit pathname argument to Load World.

It is unnecessary to do a Clear Machine before the Load World since this command does nothing on Ivory-based machines. On 3600 family machines, it serves a purpose. To avoid confusion and make boot files easier to understand and maintain, if Clear Machine is left in the boot file of an Ivory machine, it is ignored.

It is possible, although not recommended, to give the Enable/Disable IDS command here, but the recommended way is to do all that in your `Hello.boot` file, via Set Boot Options :IDS [Enable/Disable]. You should also use the Set Network Address commands in your `Hello.boot` file. (Note that if someone copies a boot file with an address in it to another machine, confusion will result.)

Note also that Disable IDS saves only an inconsequential amount of memory and overhead, so there is usually no advantage in doing it.

Here is the recommended sequence of commands for a Symbolics Ivory-based machine that is cold booting a world from the local disk:

```
Load World
Start
```

This loads the most recent world on your local disk, which is usually the one you want to boot.

For a netbooted machine, the recommended sequence is:

```
Netboot inc-site-genera-8-3
Start
```

Changes to Networks in Genera 8.0

Changes to Networks Since Genera 7.2

Improvements in Internet Security

Note that IP/TCP 4.4 includes this improvement which was made in IP/TCP 4.3.

Due to the heightened awareness of security after the recent "Internet Virus" attack, Symbolics has examined the security mechanisms in Genera and has made a change to IP/TCP to increase security.

Security is controlled by the SECURE-SUBNETS attribute of the site namespace object. In previous releases of IP/TCP, if there was no Internet entry in the SECURE-SUBNETS attribute of the site namespace object, then access was allowed from hosts on any Internet network. In IP/TCP 4.3 and 4.4, the default is to disallow access to Internet hosts, unless they are listed explicitly in the SECURE-SUBNETS attribute of the site object.

This is a change to the default behavior: now by default, no Internet hosts are trusted. Previously, the default was to trust all Internet hosts. This change has an important implication to existing sites that had no Internet entry in the SECURE-SUBNETS attribute of the site object; those sites must edit the site object to include this attribute, otherwise, no Internet hosts will be trusted.

(This change does not affect Chaosnet. The default for Chaosnet is to trust all hosts if there is no Chaos entry in the SECURE-SUBNETS attribute.)

Configuring the Internet Security Mechanism for Your Site

Security is controlled by the **secure-subnets** attribute of the site namespace object. The **secure-subnets** attribute can contain both an Internet entry and a Chaos entry. In order for a secure IP/TCP server to accept a connection, the address of that host or the network of that host must be listed in the Internet entry of the **secure-subnets** attribute of the site namespace object.

Sites that prefer to allow access to all Internet hosts should enter the value "ALL" in the Internet entry of the **secure-subnets** field.

For new sites, or existing sites whose site object never had an Internet entry for the **secure-subnets** attribute: when you install IP/TCP, no Internet hosts will be trusted by that machine, and secure IP/TCP servers will refuse connections from

Internet hosts. The solution is to edit the site namespace object and add a **secure-subnets** attribute whose first element is INTERNET. The second element should be "ALL" to trust all hosts (this restores your site to its previous behavior with respect to Internet security), or one or more Internet subnet addresses to explicitly state which subnets/networks should be trusted.

Here are some examples. The following entry allows access to all Internet hosts on subnets within 128.81.0.0:

```
Secure Subnets: INTERNET 128.81.0.0
```

The following entry allows access to all Internet hosts on 128.81.0.0 and to host 192.10.41.25:

```
Secure Subnets: INTERNET 128.81.0.0 192.10.41.25
```

The following entry allows access to all Internet hosts:

```
Secure Subnets: INTERNET ALL
```

New Keyword for net:define-server

The **:ascii-translation** keyword for **net:define-server** has been replaced with a new **:translation** keyword. Possible values are **nil**, **:ascii**, or **:unix**.

New Values for Machine-Type Host Object Attribute

Genera has three new values for the machine-type attribute of host objects in the namespace database. These values are: `MACIVORY`, `XL400`, and `UX400S`.

net:*emb-host* *Variable*

Specifies the host object representing the UNIX machine in which the local Ivory machine is embedded, that is, the value of this variable answers the question, "In what host am I embedded?"

embedded-in *name* *User Property*

A namespace user property for host objects. *name* is the name of the namespace object for the host in which an Ivory coprocessor is embedded. For a Symbolics UX-family machine the syntax "*embedded-host*|HOST" works in pathnames only when *embedded-host* has the EMBEDDED-IN namespace user-property.

Miscellaneous Changes

- The Dialnet address for Riverside has been changed to 16172704170.

- Genera 7.2 contained a mechanism that disabled a network and notified the user if an address resolution packet was received with a protocol address that matched the local protocol address. This mechanism ignored the opcode field of the packet, which allowed some broken packets to make the mechanism disable the network. Now, the mechanism does this if the opcode of the packet is known; otherwise, it ignores the packet.

New Features in Networks in Genera 8.0

- The Remote Procedure Call (RPC) protocol is now available on all Symbolics systems. (It is loaded as a separate system on 3600-family machines.) For more information, see the section "The Remote Procedure Call Facility".

- You can now use a Sun workstation for communicating with a Genera network. For more information, see the section "TCP and UDP Protocols Supported by SUN Computers as Servers".

Improvements to Networks in Genera 8.0

Performance Improvements to Networks in Genera 8.0

- NFS is more than twice as fast as in previous releases. Its data is now transmitted in 8K chunks, rather than 1.5K chunks.

- TCP performance has improved by 15 percent in this release, due to code optimization and array-register declarations.

Improvements to IP/TCP in Genera 8.0

- You can now transmit and receive IP broadcasts. IP broadcasting is integrated into the generic network system enabling any protocol using the UDP medium, either specifically or through the generic datagram, to broadcast. When appropriate, **net:find-paths-to-service-using-broadcast** returns Internet and Chaos paths.

  Note that IP broadcasts transmit through address `255.255.255.255`, so you should never give a host an Internet address with 255 as the host number.

- The tape service TAPE TCP RTAPE is now available. You can use TAPE TCP RTAPE instead of CHAOS RTAPE for more reliable performance.

- FTP errors that previously provided no error message now do so.

- The FTP server can now print directories that contain the character "/".

- When you write a file out to UNIX that has changed since you read it, the file on disk is no longer overwritten.

Miscellaneous Improvements to Networks in Genera 8.0

- MacIvory Ethernet hardware commits much less packet garbling in Genera 8.0.

- The X Windows client works much better in Genera 8.0.

- When connecting from a Symbolics machine to a remote Symbolics machine, the following asynchronous characters can now be sent:

  ```
  c-ABORT
  c-m-ABORT
  c-SUSPEND
  c-m-SUSPEND
  ```

- The Ethernet receiver process no longer fails trying to allocate a packet with a non-zero ref-count. This happened most frequently on XL400s with many network connections.

- When connecting to TCP hosts, the process state in the status line now includes the host name, consistent with what has always been done while connecting to Chaos hosts.

- Chaos has had an artificial restriction that did not allow Chaos addresses above #o75000 (subnet 122.). This restriction was based on the algorithm that stuffs routing information in a routing packet. Only 122 entries are allowed in a routing packet, but the entries do not have to be for contiguous subnets. You can now use Chaos addresses up to subnet 255.

- The Symbolics implementation of FTP now complies to the specification for directory creation. It remains compatible with older FTP implementations.

- You can now specify `NUMBER-OF-STOP-BITS`, `NUMBER-OF-DATA-BITS`, and `PARITY` for serial-pseudonet peripherals.

- An ICMP Port Unreachable message is now sent when a non-broadcast UDP packet is received at a port for which a UDP server is not defined.

- Interfaces are now automatically named. Named interfaces enable you to specify the name of an interface with which a network address connects. In order to associate an interface and a network address, specify the name of the interface in the third field of the network address in the namespace editor.

- The function **cl-neti::link-namespaces** enables a primary namespace server to find other namespaces not in the search rules, and to forcibly merge the networks between the two namespaces. Note that you should contact customer service before attempting to use this function. **cl-neti::link-namespaces** replaces the namespace linking functionality of **neti:find-site**.

- IP now receives IP datagrams larger than 1990 bytes automatically. The datagram received by upper layers appears as a regular packet. You can specify the maximum size received through **tcp::*ip-max-reassembly-size***. Note that the default is 10240 bytes.

- When transmitting IP datagrams larger than 1990 bytes, servers and protocols defined over a UDP medium can use the **:property** keyword for adding the property **:big-packets** (specifying the maximum packet size transmitted).

  You can allocate IP datagrams larger than 1990 bytes by hand using **neti:allocate-big-packet**. The macro **neti:allocate-big-packet** binds a special to the packet size desired when allocating packets. Additionally, **neti:without-big-packets** binds a special to **nil**, preventing the allocation of IP datagrams larger than 1990 bytes. The predicate **neti:big-packet-p** returns **t** if you pass a buffer that is an IP datagram larger than 1990 bytes.

  You can control the number of IP datagrams larger than 1990 bytes allocated using the variable **neti:*maximum-number-of-big-packets***. The big-packet allocator waits if the number of packets allocated exceeds the number specified in **neti:*maximum-number-of-big-packets***.

- You can now include data in the outgoing datagram for protocols you define over the generic datagram medium. You can include data in the outgoing datagram using the `:RFC-DATA` keyword to `NET:GET-CONNECTION-FOR-SERVICE` inside the `NET:DEFINE-PROTOCOL` form. For an example, see the definition of the `:NAMESPACE-TIMESTAMP` protocol.

- You can now use multiple Ethernet interfaces with UX-family and XL400 machines. You can use multiple interfaces and addresses by editing the address property on the host object and placing the name of the interface in the address property.

- **net:notify** (formerly **chaos:notify**) now works over the generic network system. You must add the service triple `NOTIFY CHAOS-SIMPLE NOTIFY` and/or `NOTIFY UDP NOTIFY` to the Symbolics machines at the site.

Changes to Internals in Genera 8.0

Improvements to Internals in Genera 8.0

It is no longer necessary to use **si:prepare-for-block-write** when using the **sys:%block-write** Ivory subprimitive. For compatibility, **si:prepare-for-block-write** is a macro that expands into **nil**.

Changes to Genera 8.0 for MacIvory

Incompatible Changes to Genera 8.0 for MacIvory

All Macintosh software (LIfe Support, the MacIvory Library, RPC Library, and the Genera application) is incompatible with previous versions of Genera, and Genera 8.0 is incompatible with previous versions of the Macintosh software.

Note that any of your C programs that use Ivory via RPC must be rebuilt in Genera 8.0. Also, if you have configured the Genera application via Configure MacIvory Application, you must do it again with the new one.

New MacIvory Features for Genera 8.0

- Genera now supports the Macintosh IIci and the Macintosh IIfx. Note that network performance for these new machines is significantly slower than for the Macintosh IIx.

- You can now run serial communication at full speed on the MacIvory.

- Shutting down the MacIvory now automatically quits the Genera application.

- If you specify an undefined remote application, the MacIvory now notifies you through a Macintosh notification screen and the application exits automatically.

- The following new defaults are available in the Configure Macivory Application menu:

  ° The menu defaults to the application currently running.

  ° You can now set the application memory size.

  ° You can now use this menu to change the "About Box" description.

  ° The arguments to "Start Screen" are now keywords instead of positional arguments.

Automatic downloading of fonts on the Macintosh is now available.

New Variables in MacIvory.Lib for Genera 8.0

MacIvory.lib now includes these static variables:

- RunningInBackground
- myProgramID
- ExitMacIvoryApplication
- RestartMacIvoryApplication (formerly RestartMacIvoryApplication routine)
- NoteMacIvoryStateChange

Note that ExitMacIvoryApplication and NoteMacIvoryStateChange are user defined routines. Additionally, you have to specify ExitMacIvoryApplication, RestartMacIvoryApplication, and NoteMacIvoryStateChange when writing a Macintosh application on the Ivory.

The InitMacivorySupport routine now contains a new variable: InitedMacivory.

Macintosh Software Converted to THINK C Version 4.0

The Macintosh software has been converted to THINK C 4.0, with some minor incompatible changes to the Lisp structures corresponding to the C structures in the Toolbox include file library.

New Keyword to mtb:with-resource

mtb:with-resource takes a new keyword, **:direction**, and allows a file reference number for the. **:sole-file** keyword (use this instead of a pathname if the file is already open). Values for **:direction** are **:input**, **:output**, or **:io**; **:io** is the default and is synonymous with **:output**. **:input** means open the file in read-only mode.

New Keyword to mtb:with-resource-handle

mtb:with-resource-handle takes a new keyword, **:direction**, and allows a file reference number for the **:sole-file** keyword (use this instead of a pathname if the file is already open). Values for **:direction** are **:input**, **:output**, or **:io**; **:io** is the default and is synonymous with **:output**. **:input** means open the file in read-only mode.

New Functions for Creating Resources

mtb:using-scratch-resfile *()* &body *body* *Macro*

Runs *body* using a scratch resource file as the current resource file. The scratch file is deleted when you quit the Genera icon.

mtb:download-resource *res-type res-name res-id res-attr data* *Function*

Creates a Macintosh resource in a scratch resource file. The first four arguments define the resource. *data* is the contents of the resource. **mtb:download-resource** returns two values: a handle and *res-id*. When you specify **nil** as *res-id*'s value, **mtb:download-resource** generates a unique id for the resource.

New Feature: MacIvory Interface to the Notification Manager Implemented

This release of Genera includes a MacIvory interface to the Macintosh OS Notification Manager, described in Macintosh Technical Note #184.

Lisp functions in the **mtb** package:

Lisp function	*Arguments*	*Value Returned*
_NMInstall	*(nmReqPtr QElemPtr)*	*OSErr*
_NMRemove	*(nmReqPtr QelemPtr)*	*OsErr*

Note that you should use extreme caution when using this feature.

Controlling the Size and Placement of a Genera Screen on a MacIvory

You can use Backup screens by using the Enable/Disable Backup screen option from the Macintosh Options menu. This option enables you to place one screen on top of another and to use each screen individually.

You can use Movable screens by using the Enable/Disable Movable screen option from the Macintosh Options menu. This option enables you to adjust the size of the Genera screen in the same manner that you adjust the size of a Macintosh screen.

Improvements to Using the Symbolics Keyboard with MacIvory

Use of the Symbolics keyboard with the MacIvory is provided primarily for compatibility with Genera applications. In addition, Symbolics supports the use of the Symbolics keyboard with native Macintosh applications.

Mapping of Apple Key Functions to the Symbolics Keyboard

- All "ordinary" characters (printing graphics, TAB, SPACE, RETURN) work normally.

- Apple's ESC key is entered using our ESCAPE key.

- Apple's COMMAND key is entered using either of our SUPER keys.

- Apple's OPTION key is entered using either of our META keys.

- Apple's CONTROL key is entered using either of our CONTROL keys.
- Apple's ENTER key is entered using our SCROLL key.
- Apple's CLEAR key is entered using our CLEAR INPUT key.
- Apple's HELP key is entered using our HELP key.
- Apple's END key is entered using our END key.
- Function keys (F1 - F15) are entered using the FUNCTION key as a shift key as follows:

Apple	*Symbolics*	*Apple*	*Symbolics*	*Apple*	*Symbolics*
F1	FUNCTION-1	F6	FUNCTION-6	F11	FUNCTION--
F2	FUNCTION-2	F7	FUNCTION-7	F12	FUNCTION-=
F3	FUNCTION-3	F8	FUNCTION-8	F13	FUNCTION-'
F4	FUNCTION-4	F9	FUNCTION-9	F14	FUNCTION-\
F5	FUNCTION-5	F10	FUNCTION-0	F15	FUNCTION-|

- The numeric keypad keys are entered using the SYMBOL key as a shift key as follows:

Apple	*Symbolics*	*Apple*	*Symbolics*	*Apple*	*Symbolics*
0	SYMBOL-0	6	SYMBOL-6	*	*see note.*
1	SYMBOL-1	7	SYMBOL-7	-	SYMBOL--
2	SYMBOL-2	8	*see note*		*see note*
3	SYMBOL-3	9	SYMBOL-9	.	SYMBOL-.
4	SYMBOL-4	=	*see note*		
5	SYMBOL-5	/	SYMBOL-/		

Note: In the table above, no mappings are provided for numeric 8, numeric =, numeric *, and numeric +. These four characters require special treatment because * is SHIFT 8 and + is SHIFT =. Symbolics distinguishes NUMERIC SHIFT 8 from NUMERIC * by enabling you to use the SHIFT key on the same side as the SYMBOL for producing the unshifted key with a SHIFT modifier, and using the SHIFT and SYMBOL keys on opposite sides for producing the shifted key without a modifier. Note that both SHIFT keys are required to produce a shifted key with a SHIFT modifier. For example:

LEFT SYMBOL 8 produces NUMERIC 8
LEFT SYMBOL LEFT SHIFT 8 produces NUMERIC SHIFT 8
LEFT SYMBOL RIGHT SHIFT 8 produces NUMERIC *
LEFT SYMBOL LEFT SHIFT RIGHT SHIFT 8 produces NUMERIC SHIFT *
LEFT SYMBOL = produces NUMERIC =
LEFT SYMBOL LEFT SHIFT = produces NUMERIC SHIFT =
LEFT SYMBOL RIGHT SHIFT = produces NUMERIC +
LEFT SYMBOL LEFT SHIFT RIGHT SHIFT = produces NUMERIC SHIFT +

The remaining keys on the extended keyboard are entered using the SYMBOL key as a shift key as follows:

Apple	*Symbolics*	*Apple*	*Symbolics*
↑	SYMBOL-i	HOME	SYMBOL-k
↓	SYMBOL-,	PAGE UP	SYMBOL-PAGE
←	SYMBOL-j	PAGE DOWN	PAGE
→	SYMBOL-l	DEL FWD	SYMBOL-RUBOUT

Interoperability of the Symbolics Keyboard and Popular Macintosh Software

Certain popular Macintosh software does not interoperate fully with the Symbolics keyboard. In some cases, a simple workaround (for example, renaming a file) exists which enables full function. In other cases, there is no workaround but the limitations are well known and are presented here for your convenience.

The Symbolics Keyboard and CloseView

CloseView, from Apple, allows the visually handicapped to magnify portions of their screen to make reading the screen easier. When using a Symbolics keyboard, however, the keystrokes used to raise and lower the magnification factor, Command-Option-↑ and Command-Option-↓, respectively, can not be entered using the techniques described in "Mapping of Apple Key Functions to the Symbolics Keyboard". Instead, you must disable magnification using Command-Option-X, open the CloseView control panel, change the magnification factor by clicking on the arrow buttons, close the CloseView control panel, and re-enable magnification using Command-Option-X.

The Symbolics Keyboard and Pyro!

Pyro!, from Fifth Generation Systems, is one of the most popular screen saver facilities for the Macintosh. Once running, Pyro! waits for you to press any key or move the mouse as an indication that you wish to restore the normal screen contents. However, when using a Symbolics keyboard, Pyro! will not deactivate itself if you press any of the righthand modifier keys (that is, SHIFT, CONTROL, META, and SUPER). Just use the lefhand modifier keys instead and Pyro! will deactivate.

The Symbolics Keyboard and QuickKeys 2

QuickKeys 2, from CE Software, is one of the most popular keyboard macro facilities for the Macintosh. For proper operation, the Symbolics keyboard software must load after MacIvory's support software but before QuickKeys 2. If you have INIT-Picker, or similar software for controlling the order in which INITs are loaded, see the documentation on said software for details on how to arrange the proper loading order. If you do not have such software, you can use the fact that the Macintosh System Software loads INITs in alphabetical order to get the desired effect. In particular, you can rename either the Symbolics Keyboard INIT or the Quick-

Keys 2 INIT to obtain the proper loading order. (We recommend that you rename Symbolics Keyboard to MSymbolics Keyboard.)

Changes and Improvements to the MacIvory Toolbox for Genera 8.0

Incompatible Changes to Low-Level Interface Routines to Macintosh Toolbox

The Lisp functions **mtb:_pbhopen** and **mtb:_pbhopenrf**, which provide an interface to the Macintosh File Manager, take a new argument called *permission* that corresponds to the Macintosh ioPermssn argument. The ioPermssn argument is used to say things such as whether to open the file read only or read/write. *permission*, a small integer, is commonly specified with something like *(cconstant fs-RdPerm)*. See *Inside Macintosh*, IV-120. Note that opening a Macintosh file with write permission alters its modification date, even when nothing is written to that file.

The new description for these functions is:

Lisp function	*Arguments*	*Value Returned*
_PBHOpen	(*name vrefnum permission dirid*)	(*refnum*)
_PBHOpenRF	(*name vrefnum permission dirid*)	(*refnum*)

Changes to MacIvory Toolbox Macros

mtb:with-open-refnum now takes these additional keyword options: **:permission**, **:dirid**, and **:resource-fork**.

The new description for this function is:

mtb:with-open-refnum *(var pathn* &key *:vrefnum :dirid :permission :resource-fork)* &body *body* *Macro*

Opens the host file named by the Lisp pathname *pathn* for the duration of *body*, binding *var* to the resulting refnum. Used when a MacOS refnum is needed instead of a Lisp stream (in contrast to **with-open-file**). The **:permission**, **:dirid**, and **:vrefnum** arguments are passed to the Macintosh File Manager (**mtb:_phbopen**) to open the file. If **:resource-fork** (default **nil**) is true, a stream to the file's resource fork is opened instead of the data fork. (PBHOpenRF vs. PBHOpen). If a Macintosh operating system error occurs while attempting to open the file, an appropriate Lisp condition is signaled.

New Toolbox Functions and Macros

mtb:mac-pathname-from-parts *vrefnum dirid name* *Function*

Given pieces of a Macintosh pathname, constructs a Lisp pathname. Recursively looks up names of parent folders until reaching the file system root.

Example:

```
(defun std-file-accept-pathname ()
  (multiple-value-bind (vrefnum name type)
      (mtb:ez-sfgetfile :types '("TEXT" "PICT"))
    (values (mtb::mac-pathname-from-parts vrefnum 0 name) type)))
```

mtb:with-mac-file-struct *(var octet-structure-type filepos refnum)* &body *body* *Macro*

Binds *var* to an octet structure of the named type read at *filepos* from the Macintosh file open on refnum.

Example:

```
(defun size-of-pict-in-pict-file (pathname)
  (declare (values width height))
  (mtb:with-open-refnum (in pathname :permission (mtb:cconstant fsRdPerm))
    (mtb::with-mac-file-struct (pict mtb::picture 512 in)
      (let ((rect (mtb:picture-picframe pict 0)))
        (values (- (mtb:rect-right rect 0) (mtb:rect-left rect 0))
                (- (mtb:rect-bottom rect 0) (mtb:rect-top rect 0)))))))
```

mtb:with-mac-file-bytes *(var len filepos refnum)* &body *body* *Macro*

Binds *var* to an octet vector of length *len*, read at filepos from the Macintosh file open on refnum.

mtb:do-mac-volumes *(volume-name volume-refnum* &optional *volumeparam)* &body *body* *Macro*

Iterates over all mounted MacFS volume structures. Body receives the *volume-name* string (without trailing colon), that volume's refnum, and, if needed, a VolumeParam structure. The string containing the volume name, and the VolumeParam structure, are allocated on the data stack, so they must be copied if they are to be used later.

Example:

```
(mtb::do-mac-volumes (name refnum) (print (list name refnum))) ==>
   ("Mac-4" -1)
   ("DSK" -2)
   ("SneakerNet Packet Floppy" -3)
```

mtb:_debugger &key *:transport-agent* *Function*

Enters the Macintosh debugger (Macsbug, for example), after halting the Macintosh at an instruction in the RPC remote entry.

mtb:_debugstr *string* &key *:transport-agent* *Function*

Enters the Macintosh debugger (Macsbug, for example), after halting the Macintosh at an instruction in the RPC remote entry, passing it *string* to print as part of its greeting.

New Features: Launching Macintosh Applications From Genera

The following functions are provided for launching Macintosh applications from Genera:

mtb:launch-mac-application *application-pathname* &rest *document-pathnames* *Function*

Launches the Macintosh application in *application-pathname*, telling it to open the documents in *document-pathnames*. If *application-pathname* is **nil**, locates the responsible application by looking up the file types of document-pathnames in the Finder's desktop file.

The returned value is a Macintosh process-id, used, for example, to inform a shell that a child has died. See Macintosh Technical Note #205 for further information on this topic.

Example:

```
(mtb::launch-mac-application
  () #p"HOST:DSK:MacIvory Applications:HyperIvory Stack")
```

Notes: If the launch does not succeed for some reason (file not found, not enough memory), the error code returned by _launch itself (and signalled by **mtb:launch-mac-application**) does not always appear to be meaningful.

In Macintosh System 7.0, Apple plans to reimplement the Macintosh software used to launch applications. This function may be made obsolete by those changes.

New Information on Numbering RPC Modules

See the file SYS:EMBEDDING;RPC;ASSIGNED-NUMBERS.TEXT for new information on assigning numbers to remote modules. We strongly recommend that you assign new numbers to existing modules, based on the information in the referenced file.

Changes to Genera 8.0 for the Symbolics UX-Family

New Features for the Symbolics UX-Family in Genera 8.0

New Value to :direction Option to File-opening Options

The **:direction** option takes a new value, **:block-input**. It can be used to open a FEP file for a disk that is mounted read-only. **:block-input** is similar to **:block**, a special case of the **:output** value for **:direction** used with the FEP file system.

New Capability for the `create-fep-partition` Program

The `create-fep-partition` UNIX program can now access remote tape drives.

For more information, see the section "create-fep-partition UNIX Program".

Applications Can Use RPC Error Reporting in Genera 8.0

In Genera 8.0, RPC contains new features that allow you to incorporate RPC error reporting into your applications. For more information, see the section "RPC Error Reporting on the Symbolics UX".

Improvements to the Symbolics UX-Family in Genera 8.0

In Genera 7.4 Ivory ECO #3, the first release for the Symbolics UX-family machine, a bug in the UNIX `rmt` server prevented it from releasing control of the tape device and exiting when the connection was closed. This bug has been fixed.

If the `genera` program is running on the UNIX in which the UX board is embedded, run bars now appear at the bottom of the UXs status line display, as on Symbolics workstations.

The Genera FBAND network used by the Copy World command now works on UXs embedded in hosts running SunOS 4.0.3.

You can now use the NFS client to communicate with an NFS server that has IP checksumming enabled.

Symbolics RPC now works over UDP.

The UX-family machine supports multiple Ethernet interfaces.

Serial is now supported; that is, **si:make-serial-stream** now works on the Symbolics UX.

A bug in the FEP Continue command has been fixed. Previously, that command could fail, reporting something to the effect that the system is not one of MacIvory or XL400.

Mouse double-clicking now works.

Update to Documentation for the Symbolics UX-Family

Support for SunOS

Genera 8.0 supports SunOS 4.0.3 and 4.0.3c.

Using the Symbolics UX-family machine as a Server

To run the Print Spooler or Dialnet on a Symbolics UX-Family machine you must declare it to be a server machine. You do this by setting the **server-machine** attribute in the machine's namespace object to Yes.

Using the :patch-atom feature of SCT

If you use the **:patch-atom** feature of SCT, the supplied string should be in "interchange case", which is to say ALL UPPERCASE LIKE THIS, so the resulting patch directories and files will be usable on a UNIX SYS host.

Changes to NFS in Genera 8.0

New Host User Property: UNIX-VERSIONS-ENABLED

Overrides (on a per-host basis) the value set by **neti:enable-unix-versions**. Values are:

nil	Specifies no backup versions of UNIX files.
t	Specifies that backup versions of UNIX files are made for .sct directories only. This is the default.
:all	Specifies that backup versions are made for all UNIX files.

neti:enable-unix-versions &optional *(value* **t***)* *Function*

Controls whether backup versions of UNIX files are created. By default, these are created only for .sct directories. Values are:

nil	Specifies no backup versions of UNIX files.
t	Specifies backup versions of UNIX files for .sct directories only.
:all	Specifies that backup versions are made for all UNIX files.

Changes to Network RPC in Genera 8.0

Network RPC includes these error conditions:

rpc:udp-rpc-connection-error *Flavor*

Signaled when a network error occurrs for the UDP connection.

Built on **sys:connection-error** and **rpc:rpc-error**.

rpc:udp-rpc-host-not-responding *Flavor*

Signaled if no reply is received for a UDP RPC request within ***udp-rpc-call-timeout***.

Built on **sys:host-not-responding-during-connection** and **rpc:rpc-error**.

rpc:tcp-rpc-stream-closed *Flavor*

Signaled when the TCP connection to the RPC server closes for any reason during the execution of an RPC call.

Built on the errors **sys:network-stream-closed** and **rpc:rpc-error**.

Changes to X Windows in Genera 8.0

Keyboard Control Facility

The keyboard control facility can be used with the Symbolics UX-family machine. To access it, type `Select Activity Keyboard Control` from a Genera Listener screen.

X Window Cut Buffers Supported for the Symbolics UX-family

X cut buffers can be accessed by means of the Cut, Paste, and Copy keys on Sun keyboards.

Miscellaneous Changes to X Windows

- Mouse tracking performance over Genera scroll bars using X has been improved. The improvement is especially significant on color systems.

- M.I.T. has fixed a problem in its X window managers, in which `uwm` and `twm` (perhaps others) did not deiconify a window when it asked to be Mapped. In the past, this problem could cause Genera to appear to freeze and the console to beep. This problem no longer occurs.

- On Symbolics UX-family machines, an X problem has been fixed. The default X display for the `-display` option of the `genera` program is now the value of the `DISPLAY` environment variable.

New Information on Numbering RPC Modules

See the file SYS:EMBEDDING;RPC;ASSIGNED-NUMBERS.TEXT for new information on assigning numbers to remote modules. We strongly recommend that you assign new numbers to existing modules, based on the information in the referenced file.

Changes to the Genera Documentation Set in Genera 8.0

Genera 8.0 includes new hardcopy and online versions of all Genera, layered product, and Ivory system documentation. Highlights of the new Genera documentation include:

- A new tutorial in Genera programming techniques, *Program Development Tutorial.* It replaces the "Program Development Tools and Techniques" tutorial that was located in Book 4 of the Genera 7 documentation set.
- A comprehensive new system index that includes entries for all Genera and Ivory system documents.
- Sections relocated to different printed documents to facilitate the lookup of related topics.

The new Genera documentation set has changed in the following ways:

- The books have been divided into User and Reference modules.
- All books from the Genera 7 documentation set have been renumbered, and new books have been added.

In the discussion that follows, Genera 8.0 book numbers and mnemonics are in bold, and Genera 7 book numbers are in plain type.

User Module

The User Module contains the following documents. Note that the Genera 7 book number or mnemonic is in the first column, with the new number or mnemonic in the second column. An asterisk indicates a book that is new in Genera 8.0.

7.0	8.0	Title
1A	**1**	*Genera Concepts*
1B	**2**	*Genera Workbook*
1C	**3**	*Genera User's Guide*
1D	**4**	*Genera Handbook*
3&6	**5**	*Editing and Mail*
none	**6**	*Program Development Tutorial**
0	**SITE**	*Site Operations*
none	**UP**	*Upgrading to Genera 8.0**
RN	**RN**	*Genera 8.0 Release Notes*
SIG	**IG**	*Genera 8.0 Software Installation Guide*
10	**IX**	*System Index*
Cards	*none*	*Reference Cards*

Reference Module

The Reference Module contains the following documents.

Old	New	Title
2A	**7**	*Symbolics Common Lisp Language Concepts*
2A	**8**	*Symbolics Common Lisp Programming Constructs**
2B	**9**	*Symbolics Common Lisp Dictionary*
5	*none*	[contents dispersed]
7A	**10**	*Programming the User Interface*
7B	**11**	*User Interface Dictionary*
4	**12**	*Program Development Utilities*
9	**13**	*Networks*
8	**14**	*Internals*
none	*none*	*Object-Oriented Programming in Common Lisp*

Note that *Object-Oriented Programming in Common Lisp* is not part of Symbolics' technical documentation, and is not available online in Document Examiner.

Contents of Reorganized and New Books

Program Development Tutorial (Book **6**) contains the new programming tutorial documentation. It takes an introductory/overview/tutorial approach to SCL and Genera.

Symbolics Common Lisp Language Concepts (Book **7**) contains roughly the first half of old Book 2A, plus some material from the second half of old Book 2A. Contents include: "Notation Conventions", "Data Types and Type Specifiers", "Numbers", "Symbols, Keywords, and Variables", "Lists", "Arrays", "Sequences", "Characters", "Strings", "Table Management", "Functions and Dynamic Closures", "Predicates", "Inline Functions and Macros", "Evaluation", "Scoping", "Flow of Control", "Declarations", and "Understanding Compatibility Issues".

Symbolics Common Lisp Programming Constructs (Book **8**) contains roughly the second half of old Book 2A, minus some miscellaneous chapters that were put into Book 8, plus CLOS and sections of old Book 5 on streams, printed representation, formatted output, and the reader. Contents include: "Structure Macros", "Flavors", "CLOS", "Conditions", "Packages", "Streams", "How the Reader Works", "Input Functions", "Printed Representation", "Output Functions", "Formatted Output", and "Writing Programs that use Magnetic Tape".

Symbolics Common Lisp Dictionary (Book **9**) contains dictionary entries for both Books 7 and 8.

Program Development Utilities (Book **12**) contains the reference material from old Book 4, plus new material on Metering, and material on Files and Pathnames from old Book 5. Contents include: "System Construction Tool", "Compiler", "Debugger", "Miscellaneous Debugging Aids", "Metering", "Naming of Files", "File and Directory Access", and "Formatting Lisp Code".

Miscellaneous Changes

"The Input Editor Program Interface" moves from old Book 4 to the *User Interface* books (Books **10** and **11**).

"Putting Data in Compiled Code Files" moves from being a chapter in old Book 4 to being a section in the Compiler chapter.

Printer documentation has been completely updated and moved to the Genera Documentation Set. For more information, see the section "Printers".

Complete documentation is now available on the IFEP Debugger. See the section "Debugging on Ivory-based Machines".

Notes and Clarifications in Genera 8.0

Notes and Clarifications Since Genera 7.2

- Much of the new Namespace Editor documentation that appeared in the *Genera 7.2 Patch Notes* was not shipped with the online documentation that accompanied Genera 7.2. The missing online documentation is shipped with Genera 8.0.

- The **format:output** function was removed from the system as of Genera 7.0. Documentation in earlier releases did not mention this.

- Both Peek's help screen and the documentation for its `Z` command stated that, by default, the Peek display was updated every two seconds. In fact, the default update interval has always been every twenty seconds.

- The documentation for the :Include Components keyword to the Compile System command was incorrect. It stated that the keyword controlled whether to compile any component systems. In fact, the keyword controls whether to load those systems. Autonomous component systems must always be compiled, if desired, explicitly.

The documentation has been corrected. For more information, see the section "Compile System Command".

- On Ivory-based systems, **copy-array-portion** is faster than **sys:vector-bitblt**; on 3600-based systems, they are about the same. If you are using an Ivory-based system and are only copying, use **copy-array-portion**.

Loading Embedding Support Systems Into a 3600-Family World

It is possible to load embedding support systems into a world running on a 3600-family machine. These systems include the following; dependencies are as shown, with the bottom system loaded first:

```
MacIvory Development           UX Development
        |                             |
MacIvory Support               RPC Development
        |                             |
Embedding Support                UX Support
        |                             |
       RPC                     Embedding Support
                                      |
                                     RPC
```

Binaries for these systems were shipped to you on the "Genera 8.0 Sources" tape.

Loading Systems Into Delivery Worlds

You can load the following systems into UX and MacIvory Delivery Worlds; dependencies are as shown, with the bottom system loaded first:

```
Hardcopy
   |
 Serial
   |
 Zwei
```

Binaries for these systems were shipped to you on the "Genera 8.0 Sources" tape.

Why Machines Sometimes Crash During Garbage Collection

Show GC Status says that you have 200,000 words of space before Start GC :Immediately will run out of space, so why does your machine crash?

The estimate of how much space the Garbage Collector is going to need is just that, an estimate. If you have not done a garbage collection before in this boot session, and if you boot a world from your local FEP, it is difficult to estimate how many pages of swap space will be used by the garbage collector due to the effects of swap migration on how much copy space is needed.

For the details of swap migration, see the section "Swap Migration and Garbage Collection".

The problem was not introduced in Genera 8.0. This has been a problem on every 3600 and Ivory release by Symbolics, but it is just now being documented.

Defining the Site Dialogue for a Delivery UX-Family Machine

When defining the site dialogue for a delivery UX-family machine, the Define Site and Set Site commands disable the console. You can restart the console by following these steps:

1. Kill the Genera program by pressing `c-C`.

2. Restart the Genera program by typing `genera`.

Correction to Conversion Tools Documentation

The last sentence in the section "Loading the Conversion Tools" should read as follows:

> For your convenience, the distribution tape also includes two sample programs you can use to try out the Conversion Tools: SYS:CONVERSION-TOOLS;CONVERSION-TEST-PROGRAM.LISP (the main example) and SYS:CONVERSION-TOOLS;CONVERSION-OCTAL-TEST-PROGRAM.LISP (for radix conversion).

The file CONVERSION-TEST-PROGRAM-5.LISP, mentioned in the documentation, is not included on the distribution tape.

Running MACSYMA in Genera 8.0

The current Genera version of MACSYMA, MACSYMA 416.15, is compatible with and runs in Genera 8.0.

On 3600-family machines, there are two minor anomalies to notice:

1. You get three warnings during the loading of MACSYMA-4:SYSTEM;SUPRV.BIN having to do with the constant **internal-time-units-per-second**. These warnings can be ignored.

2. When you click on [Hardcopy] in the Macsyma Plotting Window, it gives an error:

   ```
   The function HCI::KBD-HARDCOPY-SCREEN-WITHOUT-STATUS is undefined.
   ```

   You can press `FUNCTION Q` or `FUNCTION 0 Q` instead to get the same functionality.

On Ivory-based machines, there are no known problems. MACSYMA loads without any warnings and [Hardcopy] in the plotting window works as expected.