SECOM-L

Version 214

Reference Manual

01. March 2007

greenHouse
Software & Consulting
Karl-Heinz Weber
Heinrichstraße 12
D-45711 Datteln/Horneburg

Trademarks or Service Marks

The following are trademarks or service marks of Tandem Computers Incorporated:
Atalla, Challenge/Response, Enform, Expand, Guardian, Guardian90, Inspect, Multilan, NonStop, TACL, Tandem.
All brand names and product names are trademarks or registered trademarks of their respective companies.

The following are trademarks or service marks of *GreenHouse Software & Consulting*:
$ARROW, $AS, CRYSTAL, CURIOUS, FTPSERV-E, FUNCTRAC, MPWD, MPWD-L, PASSYNC, SECMAN, SECOM, GSTK, SSTK.

The following are trademarks or service marks of Jelinek EDV:
SECMAIN

Copyright

Copyright © 2007 by *GreenHouse Software & Consulting*. All rights reserved.
No part of this document may be reproduced in any form, including photo copying or translation to another language, without prior written consent of *GreenHouse Software & Consulting*.
Printed in Germany.

Please Comment

If you have questions or problems concerning the content of this document, please let me know! Send your comments to:

GreenHouse Software & Consulting
Karl-Heinz Weber
Heinrichstraße 12
D-45711 Datteln/Horneburg
Germany
Phone +49 (0)2363 72566
Fax +49 (0)2363 66106
Mobile +49 (0)172 23 18248
E-Mail: Info@GreenHouse.de
Internet: www.GreenHouse.de
PGP fingerprint: 3A32 D90A D125 5418
1150 2484 6629 2DD2

SECOM-L

Secure Command Manager - Lite
01. March 2007
Version 214

The Secure Command Manager (SECOM) product from GreenHouse is the most complete command level security and ID hopping tool in the NSK world. It allows an authorized user to access any resource, running with any ID, WITHOUT the need to know the IDs password.
It can be configured to control the users input to ensure, that only a subset of commands, available in a resource, can be executed (command level security).
SECOM also covers a complete access as well as management logging, and the ability to capture the session I/O (input as well as output [interactive as well as block mode], and OSS sessions).

The SECOM-L product is a lite version of the 'fully blown' SECOM product, offering these functions:

- An easy command management maintenance system:
  All SECOM-L commands are small EDIT type files.
  The command maintenance can be done using EDIT and TEDIT.
- A secure way to authorize a SECOM-L command.
  The EDIT type file has to be licensed by the users, which is defined as ID, the command resource has to run with.
  The LICENSE command is a function of SECOM-L.
- A secure platform to control users, authorized to execute SECOM-L commands.
  Access to command files is controlled by GUARDIAN, and optional SAFEGUARD.
  To execute a SECOM-L command, a user must have READ access on the command file.
- A secure platform to execute these commands.
  SECOM-L runs PRIV code. To introduce it to the system, SUPER.SUPER must be used to
  FUP LICENSE the SECOM-L program.
- The logging of all security relevant actions.
  All SECOM-L LICENSE and EXECUTION commands are logged in an entry sequenced file, which easily can be listed by ENFORM.

SECOM-L is designed to run on a cold loaded system. It does NOT need any subsystem to run.

As the name already says: SECOM-L does not have all features offered by the SECOM product, but a limited subset of functions. This keeps its
- management simple, its
- use straight forward, and
- allows an easy migration to SECOM.

A complete comparison of functions, provided by SECOM in contracts to SECOM-L, can be found at the end of this document.

SECOM-L is a GUARDIAN based security system. It is available as non native (file code 100) as well as native (file code 700 and 800) program.

SECOM-L is delivered along with the source of the log file DDL, a few command examples, and the program itself.

SECOM-L does NOT bypass any GUARDIAN or SAFEGUARD security rules!

Installation

SECOM-L comes as a self extracting PAK type file, along with the required LicenseToken, which is in a password protected normal PAK type file.

1. Upload the SECOMLP PAK type file in BINARY mode onto your Tandem system into an empty location, and name it e.g. SECOMLP.
   Make sure, the file code of SECOMLP is set to 100. In case it is not, use the

   ```
   FUP ALTER SECOMLP,CODE 100
   ```

   command to correct it.

2. This step requires, that you are logged on to SUPER.SUPER: SECOM-L runs PRIV code and needs to become licensed.
   Get to a TACL prompt, logon to SUPER.SUPER, volume over to the location, into which SECOMLP was transferred, and execute the self extracting SECOMLP PAK file:

   ```
   [run] SECOMLP
   ```

   This causes SECOMLP to:
   - extract its contents into the location, where it resides
   - run the DDL compiler to create a dictionary
   - run FUP to secure all files in the location to the best GUARDIAN settings
   - run FUP to license those files, needing to become licensed
   - run FUP to give all files to SUPER.SUPER
   - Adjusts the contents of the example command files, delivered with SECOM-L

3. Transfer the file <company-token> (e.g. GHSTOK) in BINARY mode into the same location, in which SECOMLP was loaded, and name the file: TOKEN.
   Make sure, the file code of TOKEN is set to 1729. In case it is not, use the

   ```
   FUP ALTER TOKEN,CODE 1729
   ```

   command to correct it.

4. Run the TACL Macro GETTOKEN:

   ```
   [run] GETTOKEN SECOML <password>
   ```

   where <password> is your company specific password as defined in the delivery package. This decodes the TOKEN file, and extracts the required LicenseToken SECOLTOK, which is needed to make SECOM-L run.

5. To finish the installation procedure, you need to license the supplied test commands by executing the following SECOM-L command:

   ```
   [run] SECOML LICENSE *
   ```

SECOM-L program execution

SECOM-L is a GUARDIAN based program. It accepts all user supplied run time parameters, such as CPU, PRI etc. BUT it does NOT propagate any of these parameters to the resource it starts!

SECOM-L does not have an interactive interface, nor can it be used as INLINE process.

Depending on the available options, user supplied command attributes and start-up parameters can be forwarded to the SECOM-L command resource.

Running SECOM-L without any parameter displays a brief help screen. For more details, please refer to the following sections.

A typical SECOM-L command looks like this:

```
SECOML VRPOC <file>
SECOML/CPU 2,PRI 180/SUPERTACL
```

SECOM-L command set

The SECOM-L program has the following hard coded commands:

- **-H[ELP] [COMMAND]**
  displays a brief help screen;
  this screen is also shown when SECOM-L is started without any parameter.
  When the keyword **COMMAND** is present, all supported SECOM-L command keywords are displayed.

- **[EXECUTE] <command [params]> [;<command [params]> ..]**
  executes one or more SECOM-L commands

- **LICENSE <command> [;<command> ..]**
  licenses one or more SECOM-L commands

- **LIST [<command> [;<command> ..]]**
  lists all SECOM-L commands, a user is allowed to execute
  In case <command> is a complete name (= no wildcards), the commands contents is displayed.

HELP

The **-HELP** command displays a brief help screen:

```
$GHS1 SECOML 11> secoml -h
SECOML (211) - T7172G06 - (18Jun2004)   System \BEECH, running NSK G06
Copyright (c) GreenHouse Software & Consulting 2004
Command syntax is:
  SECOML  -H[ELP]   [COMMAND]
          [EXECUTE] <command> [;<command> ..]
           LICENSE  <command> [;<command> ..]
           LIST    [<command> [;<command> ..]]
where

  -H[ELP]   displays this screen
            when the COMMAND keyword is present, all possible
            SECOML command key words are displayed.

  EXECUTE   enforces SECOML to execute <command>.
            The EXECUTE keyword is optional.

  LICENSE   licenses a SECOML command file.
            Only SUPER.SUPER, the group manager of the command ID,
            or the command ID can license a SECOML command file.

  LIST      lists all SECOML commands the current SECOML user is
            allowed to execute.
            A supplied fully qualified <command> causes SECOML to
            display the entire command contents.
```

```
 <command>  is the name of a SECOML command file, which is a
            disk file name.
            Non fully qualified commands are expected to reside
            in the same location as the SECOML object file.
            Multiple, semicolon separated, commands are allowed.
            GreenHouse extended wildcards are supported.
e.g.

SECOML LICENSE VPROC
SECOML EXECUTE STRTAPP1;RESTART2;STRTTMF     is identical with:
SECOML         STRTAPP1;RESTART2;STRTTMF
SECOML LIST    CARL{..n}.SECOL*;$GHS2.SECOML.*SUPER*

Keywords are NOT case sensitive, and can be used in any order.

Available options:
Empty PARAMs OK                        (63)
IN file definition in command file OK (62)
Propagates Process State Bits          (61)
TIMEFRAMEs in command file OK          (60)
USERs in command file OK               (59)
User supplied start-up parameters OK   (58)
EMS message generation active          (56)
$GHS1 SECOML 12>
```

All available options are displayed in the yellow marked part. Non licensed options are not shown.

The **-HELP COMMAND** displays all available SECOM-L command key words:

```
$GHS1 SECOML 30> secoml -help command
SECOML (211) - T7172G06 - (18Jun2004)   System \BEECH, running NSK G06
Copyright (c) GreenHouse Software & Consulting 2004
Possible SECOML command attributes:
COMMENT:     [comment]                   optional, multiple
ID:          <GURDIAN-ID>                mandatory
RESOURCE:    <object-file-name>          mandatory
PARAM:       <start-up parameter(s)>     optional
NOWAIT:      ON|OFF                      optional
DEFAULT:     <default location>          optional
IN:          <IN-file>                   optional
OUT:         <OUT-file>                  optional
TIMEFRAME:   <timeframe>                 optional, multiple
USER:        <user>                      optional, multiple
USERPARAM:   ADD|IGNORE|REPLACE|REQUIRE  optional
EMS:         OFF|INFORM|CRITICAL         optional
SECOML propagates process state bits.
$GHS1 SECOML 31>
```

Only enabled command key words are displayed.

EXECUTE

The EXECUTE command directs SECOM-L to read the given file, and to execute its contents. More than one command can be provided in a command line.

The command syntax is:

```
SECOML [EXECUTE] <command-1 [params]> [;<command-n [params]> ..]
```

The keyword EXECUTE is optional, and can be omitted.

The following users are allowed to execute a SECOM-L command:

- the command owner (= ID)
- when Option 59 is NOT present: All users with READ access of the command file
  when Option 59 IS present: All users mentioned in the command file

The EXECUTE logic finds out, if the given EDIT type file is a SECOM-L command type file. This is done by opening it, and checking the first few lines for consistency, and the LICTOK cryptogram for correctness. When the file is not a SECOM-L command type file, or when the LICTOC cryptogram is invalid, SECOM-L does not execute anything from the file.

<command> supports extended wildcards.

Depending on the purchased Options, the user can be allowed to supply run-time parameters **(params)** to the resource, when defined in the SECOM-L command file.

Params may contain these keywords, which are replaced by SECOM-L as follows:

- [SECOMLUSERNAME]
  This key word is replaced by the name of the current SECOM-L user;
  e.g. GHS.CARL, CarlWeber, etc.
- [SECOMLUSERID]
  This key word is replaced by the ID of the current SECOM-L user, e.g.
  100,5

LICENSE

Before SECOM-L is allowed to execute a command file, the ID, defined in the command file as the one to be used to execute the resource with, has to license the command file.

The license mechanism works like this:

1. The entire command file is checked for a correct syntax. In case unknown key words are detected, the user gets informed and the license operation aborts.
2. The used ID is checked with the ID of the user, issuing the LICENSE command. Only the ID, the IDs group manager, and SUPER.SUPER are allowed to license a command file.
3. In case all checks are OK, a 32 byte hash is computed and added to the command file.

The command syntax to license a SECOM-L command file is:

```
SECOML LICENSE <command-1> [;<command-n>; ..]
```

The keyword LICENSE is required.

The following users are allowed to license a SECOM-L command:

- the command owner (= ID, defined in the SECOM-L command file)
- the group manager of ID
- SUPER.SUPER

The LICENSE logic finds out, if the given EDIT type file is a SECOM-L command type file. This is done by opening it, and checking the first few lines for consistency. When the file is not a SECOM-L command type file for sure, it is not touched at all.

<command> supports GreenHouse extended wildcards.

The following is an example that demonstrates the SECOM-L licensing of all SECOM-L command, residing in the SECOM-L location:

```
$GHS1 SECOML 18> secoml license *
Command file:  $GHS1.SECOML.REFRESH  licensed OK
Command file:  $GHS1.SECOML.RESTART  licensed OK
Command file:  $GHS1.SECOML.SUPERT  licensed OK
Command file:  $GHS1.SECOML.VPROC  licensed OK
$GHS1 SECOML 19>
```

LIST

The LIST command displays all SECOM-L commands, the current SECOM-L user is allowed to execute.

The command syntax is:

```
SECOML LIST [<command-1> [;<command-n> ..]]
```

The keyword **LIST** is required.
<command> supports extended wildcards.
In case no **<command>** is supplied, SECOM-L lists all commands from its object-file location.
In case <command> is a complete name (= does not contain wild card characters), the entire command is displayed.

Commands are not displayed, when

- the file in question is not a SECOM-L command type file
- a user is not allowed to execute them, or when
- the license is invalid.

The LIST logic finds out, if the given EDIT type file is a SECOM-L type command file. This is done by opening it, and checking the first few lines for consistency. When the file is not a SECOM-L command type file for sure, it is not listed.

In case <command> is a 'complete' name (= name does not contain wild cards), the entire command is displayed to the user, e.g.:

```
$GHS1 SECOML 22> secoml list supert
COMMENT:   Starts an interactive SUPER.SUPER TACL
COMMENT:   Requires Option 63 = empty PARAM
COMMENT:   Requires Option 61 = to propagate process state bits
COMMENT:   Uses     Option 59 = to allow USERs (optional)
COMMENT:   Uses     Option 56 = to generate an EMS message (optional)
COMMENT:   Adjust the USER entry according to your needs:
COMMENT:   - change the entry and/or
COMMENT:   - add new entries

COMMENT:   When the LicenseToken is expired, this command will no longer
COMMENT:   function.

USER:      *
ID:        super.super
RESOURCE:  $system.system.tacl
EMS:       INFORM
LICENSEE:  SUPER.SUPER
LICDAT:    16Jun2004, 13:26:24
LICTOK:    B6139F1E224CC2E4B277D2463707D7FF
$GHS1 SECOML 23>
```

In case <command> does contain wildcard characters, the LIST command displays:

- The file name of the command
- The first COMMENT line of the command

A typical LIST output looks like this:

```
$GHS1 SECOML 205> SECOML list
       Command File                              Comment
-------------------------  ------------------------------------------------
--
COMMAND1                   Test of SECOM-L  command
COMMAND2                   Test of SECOM-L  command
COMMAND3                   Test of SECOM-L  command
INTEST                     Test of IN file
SUPERT                     Start a SUPER.SUPER TACL
$GHS1 SECOML 206>
```

Preparing a SECOM-L command

Beside the hard coded commands, SECOM-L can EXECUTE user defined commands.
A SECOM-L user defined command is stored in an EDIT type file of the following structure:

```
COMMENT:    Test of SECOM-L  command     optional
ID:         SA.CARL                      mandatory
RESOURCE:   $SYSTEM.SYSTEM.spoolcom      mandatory
PARAM:      spooler
                                         mandatory/configurabl
e
NOWAIT:     OFF|ON                       optional
DEFAULT:    $vol.subvol                  optional
OUT:        $sp.#ghs                     optional
```

Where:

COMMENT: defines a comment line.
The number of comments is limited to the size of the SECOM-L internal command file size, which actually is 32 Kbytes.
COMMENT lines can appear anywhere in the command file.
The first COMMENT line is displayed by the LIST command.
COMMENT lines are optional.

ID: defines the ID to be used to run the resource with.
The ID has to be a GUARDIAN ID. In case it is an Alias ID, it is automatically converted into the corresponding GUARDIAN ID.
The ID has to exist!
ID is mandatory.

RESOURCE: Is the resource (program), started by SECOM-L using ID as the CAID as well as PAID.
The resource has to be a fully qualified file name, and has to exist.
RESOURCE is mandatory.
To address a resource, residing in the current SYSnn subvol, use:

```
$SYSTEM.SYSTEM.<file>
```

and SECOM-L does the correct resolution automatically.
In case **Option 61** is active, SECOM-L propagates the

```
ALREADY LOGGED ON
```

process state, which e.g. allows the start of an already logged on TACL.

PARAM: Is a case sensitive string, sent to the resource as start-up message. PARAM is mandatory.

PARAM can be omitted, when **Option 63** is activated, allowing SECOM-L to start a resource for an interactive usage.

When **Option 58** is active, the SECOM-L command can receive a user provided start-up parameter at run time.
Place holders (%1% .. %9%) are supported with **Option 58,** when **USERPARAM** is set to: **REQUIRE** (see next chapter)

The PARAM string may contain these keywords, which are replaced by SECOM-L as follows:

- **[SECOMLUSERNAME]**
  This key word is replaced by the name of the current SECOM-L user; e.g. GHS.CARL, CarlWeber, etc.
- **[SECOMLUSERID]**
  This key word is replaced by the ID of the current SECOM-L user; e.g. 100,5, etc.

NOWAIT: Defines, if SECOM-L has to wait, until the started resource is stopped before the user is prompted again, or if the user has to be prompted immediately when the resource is started.
Two keywords are available:
OFF means: SECOM-L waits until the started resource stops, before it prompts the user again.
ON means: SECOM-L does not wait until the resource is finished, but comes back to the user immediately.
When NOWAIT is not supplied, SECOM-L defaults to: OFF
NOWAIT is optional.

DEFAULT: Defines the DEFAULT location which has to be communicated to the command resource. When missing, the actual users default location is used.
DEFAULT is optional.

OUT: Defines the OUT device, to which the resource has to ship its output.
OUT can be any device, except a terminal.
When missing, the SECOMN-L home terminal is used.
OUT is optional.

- Keywords are allowed in any order.
- Keywords are NOT case sensitive.
- Empty lines in the command file are accepted.

Beside the standard command attributes as defined above, the following command attributes are available, depending on the licensed functionality of SECOM-L:

```
IN:         $term
TIMEFRAME:  31Dec2003-21Jul2005
USER:       GHS.CARL
USERPARAM:  ADD|REPLACE|IGNORE|REQUIRE
EMS:        OFF|INFORM|CRITICAL
```

Where:

IN: Defines the IN file that has to be used by the resource.
IN has to be a disk file.
When IN is supplied, its file creation as well as last modification time stamp are taken into account to compute the LicenseToken for the command. This ensures, that any change of the IN file makes the corresponding SECOM-L command invalid.
IN is optional.
IN requires Option 62.

TIMEFRAME: Describes the time frame, in which the command can be executed. Only one time frame can be defined.
The timeframe format is: <start-date>[-]<end-date>, where a date is a string of the format: ddmmmyyyy, e.g. 31APR2004.
A timeframe from 01 January 2004 until the end of November 2005 would be defined as:
- 01JAN200430NOV2005 or
- 01JAN2004-30NOV2005.

The timeframe string is NOT case sensitive.
TIMEFRAME is optional.
TIMEFRAME requires Option 60

USER: Defines the user(s), allowed to execute the SECOM-L command.
Multiple entries are allowed.
Users can be defined with wildcards.
Entries for GUARDIAN users are NOT case sensitive, those for Alias users are case sensitive.
When the USER option is active, but no user is defined, all users with READ access on the command file are allowed to execute the command.
USER is optional.
USER requires Option 59.

USERPARAM: Enables SECOM-L to ship user supplied start-up parameters to the resource.
Four keywords are available:

IGNORE: SECOM-L does not ship any user defined start-up parameters to the resource (default).

ADD: In case there are user supplied start-up parameters, append them to a possibly defined command parameter. When no user parameter is present, the already defined command parameter is used as it.

REPLACE: A user supplied parameter replaces an existing command parameter.
When no user parameter is present, the already defined command parameter is used as it.

REQUIRE: A users input is **required** and
- appended to the already defined command parameter, or
- used to replace configured place holders (%n%).
  Placeholders can be in the range of: %1% .. %9%

When USERPARAM is not supplied, SECOM-L defaults to: IGNORE.
USERPARAM is optional.
USERPARAM requires Option 58.

EMS: Causes SECOM-L to ship a message to the EMS system.
Three keywords are available:

OFF : No EMS message is generated (default).

INFROM: An informational message is generated.

CRITICAL: A critical message is generated.

When EMS is not supplied, SECOM-L defaults to: OFF.
EMS is optional.
EMS requires Option 56.

- Keywords are allowed in any order.
- Keywords are NOT case sensitive.
- Empty lines in the command file are accepted.

Licensing a SECOM-L command

To allow SECOM-L to execute a command, the command file (EDIT type file) has to be licensed by SECOM-L.
The LICENSE action can be performed by:

- SUPER.SUPER
- The ID, defined in the SECOM-L command to be used to execute a resource
- The group manager of the ID, mentioned in the SECOM-L command

The LICENSE command adds three lines at the bottom of the command file:

1. LICENSEE — Is the name of the user, performing the LICENSE command.
   This is the licensees GUARDIAN or Alias name.
2. LICDAT — Is the date and time, when the license was performed.
3. LICTOK — Is a 128 bit hash value, shown in hexadecimal.
   The hash takes the following command file attributes into account:
   - the command file name (complete location name)
   - the system serial number of the system where SECOM-L runs on
   - the command file contents, including LICENSEE and LICDAT
   - the command file creation timestamp
   This mechanism ensures, that a licensed file can not be exported to a foreign system, causing a security breach, nor be moved, nor introduced by RESTORE.

   In case **Option Bit 62** is active, and IN is defined, the following two time stamps of the IN file are taken into account as well:
   - IN file creation time
   - IN file last modification time
   This assures, that the IN file can NOT be changed without making the related command license invalid.

Licensing the file shown above results in something like the following:

```
COMMENT:  Test of SECOM-L  command
ID:       GHS.CARL
RESOURCE: $SYSTEM.SYSTEM.spoolcom
PARAM:    spooler
NOWAIT:   OFF
LICENSEE: GHS.CARL
LICDAT:   31Mar2004, 15:42:29
LICTOK:   67494C2A92012DB7C42779C5F83BB458
```

Any command file change such as:

- replacing the file (different location)
- renaming the file (different name)
- changing the files contents (changed last modification time stamp)

makes the hash value invalid and prevents SECOM-L from executing the file.

In case **Option Bit 62** is active (IN is taken into account), the IN files creation and last modification time stamp are also part of the LICTOK value. Any change of the IN file makes the license of the corresponding SECOM-L command file(s) invalid.

Securing a SECOM-L command

To allow only a specific user, or group of users, the execution of a SECOM-L command, the command file has to be secured by GUARDIAN and/or SAFEGUARD: READ access on the command file defines the execution right on the SECOM-L command.

When **Option Bit 59** is active, users, allowed to execute the command, can be defined within the SECOMNL command file. A USER entry in the command file has this syntax:

```
USER:    <user>
```

where **<user>** is either a complete users name, or a user name template.

- The number of USER entries is not restricted.
- USER entries allow wildcards.

When **Option Bit 59** is active, but no USER entry is defined (this is the case when purchasing this option, and having existing command files), SECOM-L allows all users, having READ access on the file, its SECOM-L execution.

Executing a SECOM-L command

To execute a SECOM-L command, the user needs two access rights:

1. Execute access on the SECOM-L program file
2. READ access on the SECOM-L type command file, or
   a USER entry in the command file (only available with **Option 59**).

The EXECUTE command is simple and looks like this:

```
[run] SECOML [EXECUTE] <command-1 [param]> [;<command-n [param]> ..]
```

where

EXECUTE	is a keyword. It can be omitted. It is there to allow a SECOM-L command named EXECUTE[1].
<command>	is the file name of a SECOM-L command file. This can be a file name template, using the GreenHouse extended wildcard support. Multiple, semicolon separated commands, are supported.
param	is available with the USERPARAM **Option 58**. This can be more than one word. All words are taken into account, until - the end of the user supplied parameter string(s) is reached - a semicolon is encountered Two keywords are supported: - [SECOMLUSERNAME] is replaced with the name of the SECOM-L user - [SECOMLUSERID] is replaced with the ID of the SECOM-L user

[1] In case it would not have been implemented, you for sure would have asked for it, right?

The execution of the SECOM-L command looks like this:

```
$GHS1 SECOML 34> SECOML infospol

SPOOLER            STATE      LOGGING FILE                LAST ERROR
$SPLS              ACTIVE     $0
$GHS1 SECOML 35>
```

In case a SECOM-L command file is invalid, or has an invalid hash value, the user gets informed about the fail reason. The command is NOT executed.

SECOM-L expects non qualified command files to reside in the same location, where the SECOM-L object file resides.
e.g. in the example above, the file INFOSPOL is expected to reside in the location, where the SECOM-L object resides.
To execute command files, residing outside the object location of SECOM-L, the command has to be qualified, e.g.

```
$GHS1 SECOML 34> SECOML $ghs1.carl.myspool
```

Start-up PARAMs
SECOM-L does NOT take user provided start-up parameters into account – until **Option 58** is purchased.
When **Option 58** is present, a SECOM-L command may look like this:

```
$GHS1 SECOML 39> SECOML vproc $system.sys01.tacl;vproc SECOML
VPROC - T9617G03 - (01 MAY 2001) SYSTEM \BEECH    Date 14 MAY 2004, 14:32:41
COPYRIGHT TANDEM COMPUTERS INCORPORATED 1986 - 2001


$SYSTEM.SYS01.TACL
     Binder timestamp:  05MAR2002 03:36:15
    Version procedure:  T9205D46^25APR02^05MAR02
           Target CPU:  UNSPECIFIED
      AXCEL timestamp:  05MAR2002 03:36:35
VPROC - T9617G03 - (01 MAY 2001) SYSTEM \BEECH    Date 14 MAY 2004, 14:32:42
COPYRIGHT TANDEM COMPUTERS INCORPORATED 1986 - 2001


$GHS1.SECOML.SECOML
     Binder timestamp:  14MAY2004 14:24:45
    Version procedure:  T7172G06_CLTOK_315_06JAN2004
    Version procedure:  T7172G06_GHSLIB197_10MAY2004
    Version procedure:  T7172G06_NEWDES304_11FEB2004
    Version procedure:  T7172G06_PWHASH202_11FEB2004
    Version procedure:  T7172G06_SECOM-L105_14MAY2004
           Target CPU:  TNS, TNS/R
      AXCEL timestamp:  14MAY2004 14:25:05
$GHS1 SECOML 40>
```

The command string directs SECOM-L, to display the VPROCs of TACL ($SYSTEM.SYS01.TACL) and SECOM-L. The two SECOM-L commands are separated by a semicolon.
This command structure (the semicolon is used as a command separator) allows the supply of more than one command string, such as:

```
SECOML SUPERFUP INFO $A.B.C,DETAIL;VPROC $a.b.c
```

The start-up message may contain these key words:

- [SECOMLUSERNAME]
  Is replaced with the name of the current SECOM-L user, e.g. GHS.CARL, Carl Weber etc.
- [SECOMLUSERID]
  Is replaced with the current SECOM-L user ID, e.g. 100,5

Multiple keywords are supported.

To allow more than eight characters for a SECOMN-L command, make an entry in your MYMACS file, such as:

```
?section supertacl macro
run $ghs1.secoml.secomln supert
```

PARAM and ASSIGN propagation
All ASSIGNs and PARAMs, propagated from the starting instance to SECOM-L, are sent to the SECOM-L started resource as well.
SECOM-L does not support command specific ASSIGNs or PARAMs

DEFNE propagation
SECOM-L is a GUARDIAN process. All DEFINEs it gets propagated from its ancestor (e.g. a TACL) are propagated to the started resource as well.

Finding/Listing a SECOM-L command

SECOM-L command files are small EDIT type files with a well defined structure.
Use the LIST command to list all SECOM-L command files in a given location.
LIST displays only those commands, the SECOM-L user is allowed to execute.

The LIST command looks like this:

```
[run] SECOML LIST [<command> [;<command>..]]
```

where

LIST	is a required keyword.
<command>	is the file name of a SECOM-L command file. This can be a file name template, supporting the GreenHouse extended wildcards. Multiple, semicolon separated, commands are supported.

e.g.

```
$GHS1 SECOML 16> secoml list
       Command File                          Comment
-------------------------  ------------------------------------------------
--
REFRESH                    REFRESH directs SCF to refresh all disk volumes
RESTART                    Restarts any server of any PATHWAY system
SUPERT                     Starts an interactive SUPER.SUPER TACL
VPROC                      Displays the VPROC of any given object file
$GHS1 SECOML 17>
```

Security System

The security system of SECOM-L is based on GUARDIAN and SAFEGUARD.
SAFEGUARD is NOT required to make use of SECOM-L.

SECOM-L program

SECOM-L is a GUARDIAN object. It comes as native (file code 700 and 800) and non native (file code 100) object file. The SECOM-L object file needs to be LICENSED by SUPER.SUPER, because it runs PRIV code.

The best security settings are:

Owner:	SUPER.SUPER, or SECURITY.ADMIN
Security:	"OOAO" or an equivalent SAFEGUARD ACL
License:	Yes
PROGID:	No
Library:	No

Executing a PROGIDed version of SECOM-L results in an ABEND.
The same is true for executing a SECOM-L object file, that has a library attached.

SECOM-L LICENSE command

Only the owner of an ID, defined in a SECOM-L command, can execute the SECOMN-L LICENSE command. This mechanism is very similar for setting the PROGID flag on an object: That can be done solely by the file owner.

SECOM-L command

A SECOM-L command is an EDIT type file.
Users, allowed to execute SECOM-L AND to read a SECOM-L command file, are allowed to execute the SECOM-L command.
READ access on the SECOM-L command files is used to gain access to SECOM-L commands.

When **Option Bit 59** is active, the user needs READ access to the SECOM-L command file as well as a USER entry in the command file (see USER).

SECOM-L LOG system

All security relevant actions of SECOM-L are logged into an entry sequenced file.
The file is named: SECOLOG0 .. SECOLOG9.
The files are owned by SUPER.SUPER and secured to "OOOO".

SECOM-L activities are written into file: SECOLOG0.
The file has a size of ~180 Mbytes.
When it runs full, it is renamed to SECOLOG1, and a new SECOLOG0 file is created.
In case SECOLOG0 runs full again,
- SECOLOG1 is renamed to SECOLOG2
- SECOLOG0 is renamed to SECOLOG1 and
- a new SECOLOG0 file is created.

This happens until there is a file named SECOLOG9. In case SECOLOG0 runs full, SECOLOG9 is deleted, and the stack of SECOLOGn-files is renamed as explained. This gives a total log file size of ~1,800 Mbytes for SECOM-L log records.

Listing the LOG file

The log files are entry sequenced files. Their structure is described in the SECOLDDL file, which also is used by the DDL compiler at installation time to create a dictionary.
This dictionary has to be used to run ENFORM against the log files.
A simple ENFORM query looks like this:

```
?assign secollog,secolog0
open secollog;

list  eventtime
      SECOM-Luser as a17
      action
      outcome
      command.id
      command.filename
      command.resource;
```

In case you need more functions

In case you need
- more functions
- the ability to build user and command groups
- to add session tracing and command level security
- and easier way of controlling your environment

consider migrating from SECOM-L to SECOM:

SECOM for sure has all functions you are looking for!

EMS Messages

SECOM-L can be forced to generate an EMS message, when a command is executed.
This feature requires **Option 56** to be active.
To enable the EMS message generation, the command in question must have an EMS entry.
Valid EMS key words are:

- OFF
  No EMS message is generated.
  This also is the default.
- INFROM
  An informational EMS message is generated.
- CRITICAL
  A critical EMS message is generated.

The generated message has this format:

```
SECOML start of: <commandfile>/<resource> as <ID> by <SECOM-L  user> [message]
```

where

<commandfile>	name of the SECOM-L command file
<resource>	disk file name of started resource
<ID>	ID, used to execute the resource with
<SECOM-L user>	user, executing the SECOM-L command
[message]	start-up message, sent to the resource (when present)

e.g.:

```
COMMENT:   Starts an interactive SUPER.SUPER TACL
COMMENT:   Requires Option 63 = empty PARAM
COMMENT:   Requires Option 61 = to propagate process state bits
COMMENT:   Used     Option 59 = to allow USERs (optional)
COMMENT:   Used     Option 56 = to generate an EMS message (optional)
COMMENT:   Adjust the USER entry according to your needs:
COMMENT:   - change the entry and/or
COMMENT:   - add new entries

USER:      *
ID:        super.super
RESOURCE:  $system.system.tacl
EMS:       INFORM
```

Executing the command above generates a message like this:

```
04-05-26 13:57:57 \BEECH.$Z076      GHS.19.200           007172 SECOML start
                                    of: $GHS1.SECOML.SUPERT/$SYSTEM.SYS01.TACL
                                    as SUPER.SUPER by SA.CARL
```

Available Options

In addition to the basic SECOM-L functions, the following optional functions are available and can be purchased:

Option Bit	Meaning
63	Allows a command definition with an empty PARAM = enables an interactive resource
62	Takes IN into account = allows the execution of OBEY type files **IN: <diskfile>**
61	Propagates logged-on and stop-at-logoff process state bits = allows the start of an already logged on TACL WITHOUT the need to know a password
60	Takes an execution time frame into account = not-to-be-used-before ... not-to-be-used-after **TAMEFRAME: <from><until>**
59	Allows the definition of users, having execution access rights on the command, in the command file **USER: <user>**
58	Enables SECOM-L to take user provided start-up parameters into account, and to ship them to the resource **USERPARAM: ADD|REPLACE|IGNORE|REQUIRE**
57	Future enhancement
56	Causes SECOM-L to generate an EMS message on commands, when defined. **EMS: INFORMATION|CRITICAL**
55	Allows SECOM-L to trace an interactive users session. The keyword RESOURCE is replaces by the key word TRACE. **TRACE: <$vol.subvol.program>** TRACETYPE controls the functionality of TRACER. **TRACETYPE: LOG|TRACE|BLOCK|OSSLOG|OSSTRACE|OSSBLOCK**
54	Enforces an inactivity timeout on an interactive session. Requires Option 55! **TIMEOUT: <minutes>**

greenHouse

SECOM-L command file summary

Keyword	Value	Meaning	Option
COMMENT	Any text; case sensitive. Multiple lines are allowed.	Describes the command. The first line is displayed with the LIST command.	Optional
RESOURCE	Fully qualified program name. $SYSTEM.SYSnn can be defined as: $SYSTEM.SYSTEM	Program to be started by SECOM-L.	Mandatory
TRACE	Fully qualified program name. $SYSTEM.SYSnn can be defined as: $SYSTEM.SYSTEM	Program to be started by SECOM-L.	Replaces RESOURCE **Option 55**
ID	Fully qualified **GUARDIAN** name. Alias names are automatically converted into the GUARDIAN equivalent.	ID to be used to start the RESOURCE with.	Mandatory
DEFAULT	Subvol ([\system.]$vol.subvol)	Default location to be shipped to the RESOURCE.	Optional
OUT	Any fully qualified file name, **except** a terminal name	Defines the OUT device for the RESOURCE.	Optional
NOWAIT	OFF (default)	OFF = SECOM waits, until the started resource stops, before it comes back to the user.	Optional
	ON	ON = SECOM-L does NOT wait until the started resource is stopped, but comes back to the user immediately.	
IN	Any fully qualified file name, except a terminal name	Defines the IN device for the resource	Optional **Option 62**
PARAM	Start-up string to be shipped to the resource.	Defines the action of the resource (start-up parameter)	Mandatory Optional with **Option 63**
	Key word: [SECOMLUSERNAME] in PARAM	Is replaced by the current SECOM-L users name, e.g. GHS.CARL	Optional
	Key Word: [SECOMLUSERID] in PARAM	Is replaced by the current SECOM-L users ID, e.g. 100,5	Optional
TIMEFRAME	<from>-<to> **12JAN2004-31DEC2004**	Defines the time frame, in which the command can be executed.	Optional **Option 60**

USER	A user name (GUARDIAN, Alias), or a user name template. Multiple lines are allowed.	When present defines the user(s), allowed to execute the command.	Optional **Option 59**
USERPARAM	IGNORE (default)	Ignores a user supplied parameter (default)	Optional **Option 58**
	REQUIRE	A user parameter is required and - appended to the already defined command parameter - used to replace place holders in the already defined command parameter(%n%) when defined	
	ADD	Adds an optional user supplied parameter to the defined command parameter(s)	
	REPLACE	Replaces command defined parameter(s) with the user supplied one	
EMS	OFF (Default)	No EMS message is generated	Optional **Option 56**
	INFORM	An informational EMS message is generated	
	CRITICAL	A critical EMS message is generated	

TRACETYPE	LOG	Terminal INPUT is recorded	Optional Requires **Option 55**
	TRACE	Terminal INOUT as well as OUTPUT is recorded	
	BLOCK	Terminal INPUT and OUTPUT as well as ALL BLOCK mode is recorded	
	OSSLOG	Terminal INPUT and OSS sessions are recorded	
	OSSTRACE	Terminal INPUT and OUTPUT and OSS sessions are recorded	
	OSSBLOCK	Terminal INPUT and OUTPUT as well as ALL BLOCK mode and OSS is recorded	
TIMEOUT	Inactivity timeout in minutes. Valid values are: 1 .. 1440	Inactivity timeout of an interactive session	Optional **Option 54** Requires **Option 55**
LICENSEE	Name of user, who performed the command licensing, e.g.: **`CarlWeber`**	Inserted at LICENSE time	N/A
LICDAT	Date of licensing, e.g.: **`23Apr2004, 10:53:54`**	Inserted at LICENSE time	N/A
LICTOK	LicenseToken, 32 byte HEX, e.g.: **`B299A6972B13A5A3953C0D32AE6C7B78`**	Inserted at LICENSE time	N/A

Keywords

- Keywords are NOT case sensitive.
- Keywords are always terminated by a colon, followed by one or more blanks.
- Keywords can appear in any order.

File names

- File names always should be fully qualified.
  Non qualified file names are resolved with the location ($vol.subvol) of the SECOM-L object file.

User names

- GUARDIAN user names, or templates, are NOT case sensitive.
- Alias user names, or templates, ARE case sensitive.

UserParam

- A case sensitive string.

Key Words

Two key words are supported:

1. [SECOMLUSERNAME]
   Is replaced with the current SECOM-L users name, e.g GHS.CARL, CarlWeber, etc.
2. [SECOMLUSERID]
   Is replaced with the current SECOM-L users ID, e.g. 100,5

Place holders (%n%)

Place holders in the range of %1% .. %9% can be defined in the PARAM line of a SECOM-L command.
When Option 58 is available, and USERPARAM is set to REQUIRE, the place holders become resolved with the user defined string(s).
When no place holders are defined, and USERPARAM is set to REQUIRE, the user parameters are appended to the string defined in PARAM.

Examples

Here are a few examples, showing how to set up a SECOM-L command file.
To make these command files work on your system., you have to license them by executing the SECOM-L LICENSE command.

```
SECOML LICENSE *
```

Because the commands run with the SUPER.SUPER ID, only SUPER.SUPER can successfully execute the SECOML LICENSE command on the following commands.

The **LICENSEE, LICDAT** and **LICTOK** entries as shown in the examples below are for sure invalid on your system, and become replaced by new values when performing the **SECOML LICENSE** command.

REFRESH

To 'refresh' the disk volumes on your system (= to update all EOFs, and disk file labels) , the

```
SCF CONTROL <$vol>,REFRESH
```

command has to be executed. To successfully do this, you have to be a member of the SUPER-Group. In case you are not – use this SECOM-L command:

```
COMMENT:    REFRESH directs SCF to refresh all disk volumes
COMMENT:    Command syntax is:
COMMENT:      SECOML REFRESH
COMMENT:    Used Option 59 = to allow USERs (optional)

USER:       *
ID:         super.super
RESOURCE:   $system.system.scf
PARAM:      control $*,refresh
```

The SECOML command to execute the REFRESH command is:

```
SECOML REFRESH
```

The delivered command file allows EVERY local user to execute the SCF REFRESH command. Please adjust the USER entry.

RESTART
The RESTART command restarts any server of any given PATHWAY system.
The delivered command file allows EVERY local user to restart any PATHWAY server.
Please adjust the USER entry.

The command syntax is:

```
SECOML RESTART <$monitor> <server>
```

e.g.

```
SECOML RESTART $GHS SESERVER
```

```
COMMENT:   Restarts any server of any PATHWAY system
COMMENT:   Command syntax is:
COMMENT:     SECOML RESTART <$pathmon> <server>
COMMENT:   Requires Option 58 = to allow USERPARAM
COMMENT:   Used     Option 59 = to allow USERs (optional)

USER:      *
ID:        super.super
RESOURCE:  $system.system.pathcom
PARAM:     %1%;freeze %2%;stop %2%;stop %2%;thaw %2%;start %2%
USERPARAM: REQUIRE
```

To restart the server SESERVER of the PATHWAY application $GHS, type:

```
SECOML RESTART $ghs seserver
```

To restrict the RESTART of all servers to e.g. the PATHWAY system $ABC, use this SECOM-L command:

```
COMMENT:   Restarts any server of any PATHWAY system
COMMENT:   Command syntax is:
COMMENT:     SECOML RESTART <server>
COMMENT:   Requires Option 58 = to allow USERPARAM
COMMENT:   Used     Option 59 = to allow USERs (optional)

USER:      *
ID:        super.super
RESOURCE:  $system.system.pathcom
PARAM:     $ABC;freeze %1%;stop %1%;stop %1%;thaw %1%;start %1%
USERPARAM: REQUIRE
```

To restart the server SESERVER, type:

```
SECOML RESTART seserver
```

SUPERT

The SUPERT command file enables the configured user, to get access to a SUPER.SUPER TACL WITHOUT the need to know the SUPER.SUPER password.

The delivered command file allows EVERY local user interactive access to SUPER.SUPER!

Before licensing this command, please adjust the USER entry to those users, you are willing to give interactive access to SUPER.SUPER!.

You can change the USER entry, and/or add new USER entries.

e.g.

```
USER:       super.*
USER:       *.MGMT
USER:       SEC.ADMIN
```

You also can delete the USER entry: In this case, all users with READ access to the command file SUPERT are allowed to get interactive access to SUPER.SUPER.

```
COMMENT:    Starts an interactive SUPER.SUPER TACL
COMMENT:    Requires Option 63 = empty PARAM
COMMENT:    Requires Option 61 = to propagate process state bits
COMMENT:    Used     Option 59 = to allow USERs (optional)
COMMENT:    Used     Option 56 = to generate an EMS message (optional)
COMMENT:    Adjust the USER entry according to your needs:
COMMENT:    - change the entry and/or
COMMENT:    - add new entries

USER:       *
ID:         super.super
RESOURCE:   $system.system.tacl
EMS:        INFORM
```

VPROC

The VPROC command runs the VRPOC program with the SUPER.SUPER ID, allowing the user to display the VPROC of any object on the system without the need to re-logon to SUPER.SUPER, or the object file owner.

The delivered command file allows EVERY local user to run the VPROC command.

Please adjust the USER entry.

```
COMMENT:    Displays the VPROC of any given object file
COMMENT:    Command syntax is:
COMMENT:      SECOML VPROC <file>;[VPROC <file>..]
COMMENT:    Requires Option 58 = to allow USERPARAM
COMMENT:    Used     Option 59 = to allow USERs (optional)

USER:       *
ID:         super.super
RESOURCE:   $system.system.vproc
USERPARAM:  REQUIRE
```

Comparison of SECOM-L / SECOM

SECOM-L is a lite version of the SECOM product. It does not offer all features of the "fully blown" SECOM product.
Below is a list of features, showing the differences between these two GreenHouse products.

Feature	SECOM-L	SECOM
Length of command name	1 .. 8 Bytes has to start with an alpha character (disk file name)	1 .. 32 Bytes (all characters)
Command execution confirmation	No	Yes
Command execution reason input	No	Yes
Command resource	Any local object	Any local and remote object
Command ID	GUARDIAN User only	GUARDIAN User Alias User Any – even non existing - ID number
Process name of resource	automatic	configurable
Process name length	fixed	configurable
NOWAIT execution	Yes	Yes
Configuration of default location	Yes	Yes
Configuration of IN	Optional	Yes
Configuration of OUT	Yes	Yes
OUT is automatically given to the SECOM user	No	Yes
Configuration of Home Terminal	No	Yes
Configuration of Library	No	Yes
Configuration of Swap file	No	Yes
Configuration of Low/High PIN	No	Yes
Terminal I/O tracing	Optional	Yes
Inactivity timeout	Optional	Yes
Pre-defined resource CPU	No	Yes
Automatic load balancing	No	Yes
Configuration of resource priority	No	Yes
Predefined start-up parameters	Yes	Yes
User defined parameters at run-time	Optional	Yes
Support of start-up message place holders (%n%)	Optional	Yes
Check of user defined run-time parameters	No	Yes
Command specific PARAMs	No	Yes
Command specific ASSIGNs	No	Yes
Default PARAM and ASSIGN shipment	Yes	Yes
DEFINE propagation	Yes	Yes
Auto command expiration	Yes	Yes
Block mode interface to commands	No	SECOMCI
OSS support	Optional	Optional
Event report to the EMS system	Optional	Yes
Command execution log	Yes	Yes
BATCH type commands (completely defined)	Yes	Yes
CHK type commands (interactive use and command contents check)	No	Yes
INT type commands (takes user parameters into account)	Optional	Yes
CON type commands (chained commands)	No	Yes
TACL-Macro and -Routine execution	No	Yes
Logging of command licensing	Yes	n/a

Logging of command management	No	Yes
Data base management by:	EDIT/TEDIT	PATHWAY
GUI for data base management	N/a	SECWIN
GUI for log analysis	No	TALIS
GUI for I/O trace analysis	Optional	TALIS
Command Authentication (Password, Challenge/Response, TimeToken)	No	Yes
Command Authorization (4 eye principle)	No	Yes
Command restriction (IP addresses, device)	No	Yes
Command restriction to a time/date	Optional	Yes
Command expiration	Optional	Yes
Local and Remote command users	SAFEGUARD	Yes
User Authentication (Password, Challenge/Response, TimeToken)	No	Yes
User expiration on SECOM command	SAFEGUARD	Yes
User restriction (IP addresses, device, date and time)	No	Yes
Command access management	Optional GUARDIAN SAFEGUARD	Yes
User group support, and group management	GUARDIAN SAFEGUARD	Yes
Command group support and management	No	Yes
Wildcard support for users	Optional	Yes
Wildcard support for user groups	N/A	Yes
Wildcard support for command groups	N/A	Yes
Interactive data base maintenance	N/a	Yes
Centralized command management	No	Yes
Centralizes command management and automated command distribution	No	Yes
Batch interface	Yes	Yes
Interactive interface	No	Yes
Can be used as INLINE process	No	Yes
Check of process stacks	No	Configurable
Finds its real creator ID (CAID)	No	Yes

In case you stumble into any glitch, or problem, please contact me immediately at:
Carl.Weber@GreenHouse.de

Carl Weber
GreenHouse Software & Consulting
01. March 2007
www.GreenHouse.de