

Support Note For Using NS AutoTMF under OSS and Java

This document provides instructions on the installation and usage of the NS AutoTMF product for OSS and Java applications to allow them to access Audited Guardian Enscribe files without the need to add TMF transaction logic to the application. It is assumed that the user has access to the AutoTMF documentation and is familiar with the principles of the product.

Currently only OSS applications developed under C and Cobol as well as Java classes using the JEnscribe EnscribeFile class have been validated for use with AutoTMF. When implementing AutoTMF under OSS on an S-series machine, it is advisable to perform a *noft listsrlfixups* and a *noft listsrlinfo* on one or more typical object files and open a case with the GMCSC asking to have the information sent to development. Due to the nature of SRLs, development support may be required for initial implementation.

NS AutoTMF Installation

Install the AutoTMF product following the normal Guardian installation instructions. The only caveat is that in order for an OSS process to access the Escort runtime libraries, the installation must be located on a non-virtual drive. This is a limitation of OSS and the Guardian File System.

Additional OSS Installation Steps

Scripts and Exports

Obtain from HP the **escortoss** shell script utility that provides access to the Escort command interpreter from OSS. Copy this file to an OSS directory and make certain that your **.profile** is updated to reflect the PATH to this tool.

Depending on your OSS environment you will need to complete and insert the following lines in either your **.profile** or in a common profile area containing directives Java developers and users:

```
export _RLD_LIB_PATH=/... <- PATH to the PREPARED libtdmext.so library
export _RLD_FIRST_LIB_PATH=/G/SYSTEM. ESCORT <- Escort installation subvol
```

Using the **escortoss** Script

The **escortoss** script is a simple shell script designed to provide access to a limited set of the functions of the Guardian Escort Command Interpreter. Executing **escortoss** with no parameters will yield the following help screen:

Usage: `escortoss <cmd> <object file>`

The OSS Escort CI supports the following commands:

```
pi          - provides program information
pi u       - provides unresolved procs information
pi x       - provides proc cross-reference information
prepare    - prepares a program and sets the DLL
```

Before using **escortoss** it must be edited to update the location of the AutoTMF installation so that it can locate the Escort CI. Because the Escort CI operates only on Guardian files, a temporary copy of an OSS object is copied to Guardian space into a file named ZZOSSTMP. Ideally this file would reside on the AutoTMF installation subvolume, but if security reasons prevent this the file may be located on any *non-virtual* drive. **Escortoss** should then be modified to reflect the path to this location. The script deletes ZZOSSTMP after each execution.

Note that the application developers will require R,W,P,C access to the Guardian ZOSSTMP file.

Using Java - Preparing the EnscribeFile Class

The key to using AutoTMF with the Java JToolkit is to PREPARE the library **libtdmext.so**. This JNI (library) contains the EnscribeFile class and performing an Escort PREPARE on this file will cause these Guardian File System calls to be intercepted by AutoTMF.

Obtain a copy of **libtdmext.so** from the JToolkit installation directory. The original release of this library is typically installed under the path **/usr/tandem/javaexth10/lib/** but may vary depending on the JToolkit release level installed at your site. Place a copy of **libtdmext.so** in the directory specified in your **export _RLD_LIB_PATH=** command. For example, if your export command is:

```
export _RLD_LIB_PATH=/home/development/java/lib
```

Place a copy of the library as follows:

```
cp /usr/tandem/javaexth10/lib/ /home/development/java/lib
```

Then use **escortoss** to PREPARE the library:

```
escortoss prepare /home/development/java/lib/libtdmext.so
```

Preparing **libtdmext.so** will cause all Guardian File System calls made by this library to be intercepted and the **ESCRUNDL** to be placed at the head of the DLL load list. Once **libtdmext.so** has been prepared, the **define =_ESCORT_MONITOR** added, and **_RLD_LIB_PATH** and **_RLD_FIRST_LIB_PATH** exported into the environment, any Java program executed will use the AutoTMF runtime library and have full use of all of its features.

TNS/R systems follow the same rules as TNS/E except that the Userlibrary will be set to **ESCRUNNM**. TNS/R DLLs are not supported by AutoTMF.

C and Cobol Applications

Unlike Java programs where all of the Guardian File System calls are isolated in a single library provided by HP, application programs developed using C or Cobol will have the File System calls embedded directly in the application object. For this reason it is necessary to perform an **escortoss** PREPARE of each executable application object prior to using it:

```
escortoss prepare /home/develop/c/exe/my_c_exe.exe
```

or

```
escortoss prepare /home/develop/cobol/my_cobol_exe.exe
```

Once the **define =_ESCORT_MONITOR** has been added the C and Cobol programs maybe executed and have full use of all of the features of AutoTMF.

Using the Escort TRACE Facility from OSS

AutoTMF includes an extensive TRACE tool that will capture all calls intercepted by AutoTMF. In order to use this tool an OSS or Java process must have a Guardian Name assigned to it. The process name must be a maximum of 4 characters in length. The following are examples for a C or Cobol application and for a Java application:

C or Cobol :

```
run -name=/G/ZTST my_c_exe.exe <- Corresponds to Guardian Process Name $ZTST
```

Java:

```
run -name=/G/ZTST java my_java.class
```

The following steps will capture a TRACE of process name \$ZTST into the Guardian file **\$system.escort.mytrace**:

- 1) **gtacl -c 'run \$system.escort.escort trace to \$system.escort.mytrace,process \$ztst;'**
- 2) **run -name=/G/ZTST java my_java.class**
- 3) **gtacl -c 'run \$system.escort.escort trace stop;'**

To view the trace:

```
gtacl -c 'fup copy $system.escort.mytrace'
```

See the AutoTMF Documentation for further TRACE options and features.

Revision History

| | |
|-----------|---|
| 28Sep2006 | First release. |
| 05May2009 | Updated to remove H-series limitation |
| 20Jul2009 | Added SRL caveat and removed monitor define |