

# Release Note

Tmax 6 Fix#1

**TMAXSOFT**

## 저작권 공지

Copyright © 2024 TmaxSoft Co., Ltd. All Rights Reserved.

## 제한된 권리

이 소프트웨어(Tmax®) 사용설명서와 프로그램은 저작권법과 국제 조약에 의해 보호됩니다. 사용설명서와 프로그램은 TmaxSoft Co., Ltd.와의 사용권 계약 하에서만 사용할 수 있으며, 사용설명서는 사용권 계약의 범위 내에서만 배포 또는 복제할 수 있습니다. 이 사용설명서의 전부 또는 일부를 TmaxSoft의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단으로 전송, 복제, 배포하거나 2차적 저작물을 작성할 수 없습니다.

이 소프트웨어 사용설명서와 프로그램의 사용권 계약은 어떠한 경우에도 사용설명서 및 프로그램과 관련된 지적 재산권(등록 여부를 불문)을 양도하는 것으로 해석되지 않으며, 브랜드나 로고, 상표 등을 사용할 권한을 부여하지 않습니다. 사용설명서는 오로지 정보 제공만을 목적으로 하며, 이로 인한 계약상의 직접적 또는 간접적 책임을 지지 않습니다. 또한 사용설명서 상의 내용이 법적 또는 상업적인 특정 조건을 만족시킬 것을 보장하지 않습니다. 사용설명서는 제품의 업그레이드나 수정에 따라 예고 없이 변경될 수 있으며, 내용상의 오류가 없음을 보장하지 않습니다.

## 상표 공지

Tmax®, Tmax WebtoB®와 JEUS®는 TmaxSoft Co., Ltd.의 등록 상표입니다. 본 사용설명서에 기재된 모든 제품과 회사 이름은 각각 해당 소유주의 상표로서 참조용으로만 사용되며 반드시 상표 표시 (™, ®)를 하지는 않습니다.

## 오픈소스 소프트웨어 공지

본 제품의 일부 파일 또는 모듈은 다음의 라이선스를 준수합니다. : openssl-0.9.7.m, zlib-1.1.4, expat-2.0.0, net-snmp, DCE1.0, pthread, google-diff-match-patch, libevent, getopt

관련 상세한 정보는 제품의 다음의 디렉터리에 기재된 사항을 참고하시기 바랍니다. :  
\${INSTALL\_PATH}/license/oss\_licenses

## 안내서 이력

제품 버전	안내서 버전	발행일	비고
Tmax 6 Fix#1	3.1.1	2024-09-24	-
Tmax 6 Fix#1	2.1.2	2019-09-11	-
Tmax 6	2.1.1	2015-07-31	-

# 목차

1. 소개	1
1.1. 릴리즈 이력	1
2. Tmax 6 Fix#1	2
2.1. 추가 기능	2
2.1.1. ENGINE	2
2.1.2. 클라이언트/서버	8
2.1.3. 유틸리티	20
2.1.4. 관리도구	22
2.1.5. TDL	26
2.1.6. 도메인 게이트웨이	27
2.1.7. Tuxedo 게이트웨이	28
2.1.8. JAVA 게이트웨이	28
2.1.9. TCP 게이트웨이	28
2.1.10. Entera 게이트웨이	28
2.1.11. RPC	29
2.1.12. Tcache	29
2.1.13. 기타	30
2.2. 변경 기능	30
2.2.1. ENGINE	31
2.2.2. 클라이언트/서버	33
2.2.3. 관리 도구	35
2.2.4. 유틸리티	35
2.2.5. TDL	36
2.2.6. Tcache	36
2.2.7. 도메인 게이트웨이	37
2.2.8. Tuxedo 게이트웨이	37
2.3. 버그 패치	37
2.3.1. ENGINE	37
2.3.2. 클라이언트 / 서버	42
2.3.3. 도메인 게이트웨이	46
2.3.4. TCP 게이트웨이	47
2.3.5. Tuxedo 게이트웨이	47
2.3.6. Java 게이트웨이	49
2.3.7. Entera 게이트웨이	51
2.3.8. TDL	51
2.3.9. 유틸리티	52
2.3.10. 관리 도구	54
2.3.11. HostLink	55
2.3.12. RPC	56

2.3.13. Tcache .....	57
2.3.14. 기타 .....	57
3. Tmax 6 .....	59
3.1. 신규 기능 .....	59
3.1.1. Tmax Deployment 기능 추가 .....	59
3.1.2. TmaxGrid 기능 추가 .....	59
3.1.3. Multiple RM에 DB2 지원 .....	59
3.1.4. 보안 기능 강화 및 유연성 제공 .....	59
3.1.5. PHP 인터페이스 제공 .....	59
3.2. 기능 개선 .....	59
3.2.1. 이름 길이 확장 .....	59
3.2.2. 시간 정밀도 개선 .....	60
3.2.3. tmbboot / tmdown 로깅 메시지 개선 .....	60
3.2.4. Tmax Log 메시지 통합 .....	61
3.2.5. Debug/Release Binary 통합 .....	61
3.2.6. clilib, svrplib에 tmaxoserror 함수 추가 .....	61
3.2.7. Svcrypt 사용자 로깅 지원 .....	62
3.2.8. 도메인 게이트 웨이의 DIRECTION 동적 변경 기능 추가 .....	62
3.2.9. 서버 프로세스들에 대한 기동시각 출력기능 추가 .....	62
3.2.10. MSGSIZEWARN, MSGSIZEMAX 옵션 추가 .....	62
3.2.11. runtime 설정 강화 .....	63
3.2.12. runtime 로그레벨 변경 강화 .....	64

# 1. 소개

본 안내서는 릴리즈 대상 제품의 새로운 기능과 이전 버전과 다른 변경된 사항을 정리한 안내서이다. Tmax는 기존 버전에 비해 다양한 실사용 환경에 적용하기 위한 기능이 추가되었다. 그러나 본 안내서는 Tmax의 전체 내용을 다루기 어려우므로 자세한 내용은 해당 안내서를 참고한다.

## 1.1. 릴리즈 이력

일자	버전
2015-07-31	Tmax 6
2019-09-11	Tmax 6 Fix#1

## 2. Tmax 6 Fix#1

본 장에서는 Tmax 6 Fix#1 릴리즈에서 추가된 새로운 기능과 변경된 기능에 대해 간략히 설명한다.

### 2.1. 추가 기능

본 절에서는 추가된 주요 기능에 대해 설명한다.

#### 2.1.1. ENGINE

- **NODE 절에 SLOG\_RNAME 기능 추가**

- NODE 절에 SLOG\_RNAME=Y/(N)를 설정할 때 서버 타입이 아니라 서버 이름으로 slog에 출력한다.
- GATEWAY, HMS, RQS, TMS 타입의 서버는 위 설정을 하지 않으면 실제 서버 이름이 아닌 서버 타입을 slog에 출력된다.
- 관련 이슈 : [IMS 111191](#)

- **TMM 비정상 종료될 때 모든 프로세스가 종료하는 기능 추가**

- DOMAIN 절 및 NODE 절에 MAC=Y/(N)를 추가한다.

MAC 옵션을 Y로 설정하면 TMM이 비정상 종료되거나 TMM과의 domain-socket 채널이 끊겼을 때 모든 프로세스들이 TMM의 연결 끊김을 감지한 이후에 강제 종료한다.

서버 프로세스들은 서비스를 수행 중이면 TMM의 종료를 즉시 감지할 수 없으며, 서비스 수행이 완료된 이후에 강제종료된다.

- 관련 이슈 : [IMS 115397](#)

- **TMMOPT -e/-o \$(CDATE) 등 매크로 기능 추가**

- TMMOPT 항목의 -e / -o 옵션에 다음의 매크로를 사용할 수 있다.
  - \$(CDATE)
  - \$(CTIME)
  - \$(CYEAR)
  - \$(CMONTH)
  - \$(CMONTHS)
  - \$(CDAY)
  - \$(CWDAY)
  - \$(CWDAYS)
  - \$(CYDAY)
  - \$(CHOUR)
  - \$(CMINUTE)

- \$(CSECOND)
- \$(PID)
- CLHOPT 항목의 -e / -o 옵션도 위와 동일하게 사용 가능하다.
- 관련 이슈 : [IMS 117565](#)

#### • CLH만의 MAC 기능 추가

- NODE 절에 MAC=N(기본값)일 경우

만약 TMM과의 연결만 끊기고 TMM 프로세스 자체는 계속 살아있는 상황이었다면 CLH는 계속 구동하면 안된다. 왜냐하면 TMM이 CLH와의 연결이 끊기는 순간 해당 CLH는 종료된 것으로 가정하고 다른 CLH를 기동할 것이기 때문이다. 그래서 CLH가 TMM과 연결이 끊겼을 때, 살아있는지 검사를 위해 TMM에게 재연결을 시도한다.

이와 관련된 로그는 다음과 같고 INFO 로그 레벨로 출력된다.

```
2011 TMM connection closed : %d
```

만약 TMM과 연결이 성공하면 TMM이 해당 CLH에게 SIGTERM 시그널을 보내 그 CLH를 종료시킨다. 이와 관련된 TMM 로그는 다음과 같다.

```
TMM3067 SIGTERM sent to disconnected CLH, pid = %d
```

- NODE 절에 MAC=Y일 경우

다른 Tmax 프로세스와 마찬가지로 TMM과의 연결이 끊기는 순간 FATAL 로그를 출력하고 종료한다. 이와 관련된 로그는 다음과 같고 FATAL 로그 레벨로 출력된다.

```
2011 TMM connection closed : %d
```

- 관련 이슈 : [IMS 115531](#)

#### • Tmax 기동 시에 CLL, CLH가 TMM에 register됐을 때 로그 추가

- CLH와 CLL이 기동 과정에서 TMM과 연결되었을 때 아래의 로그가 출력된다.

```
TMM3069 CLH (%d) registered, fd = %d, pid = %d [TMM5207]
```

```
TMM3070 CLL registered, fd = %d, pid = %d [TMM5208]
```

- 관련 이슈 : [IMS 111191](#)

#### • 라이선스 발급기에서 hostname을 256자까지 처리할 수 있도록 변경

- 환경설정 파일의 NODE 절 HOSTNAME을 256자까지 지원한다. NODENAME은 16자 제한이 그대로 있기 때문에 16자 이상의 경우 반드시 HOSTNAME을 함께 지정해야 한다.
- 관련 이슈 : [IMS 150028](#)

• **TCS 서버 타입에서 서비스 수행 시점이 아닌 대기 시점에 비정상적인 에러가 연속으로 발생하는 경우 강제 종료하는 기능 추가**

- 연속적으로 에러가 64번 발생하는 경우에 다음과 같은 에러를 출력하고 비정상 종료한다. 그 외에 정상적으로 서비스가 수행되면 누적되었던 에러 발생 카운트가 초기화된다.

```
(F) SVR3129 svr is in abnormal state [SVR0054]
```

- 관련 이슈 : [IMS 171100](#)

• **Tmax 환경설정 SERVER 절에 TPRETURN 옵션 추가**

- 서비스 타임아웃이 발생하여 tpsvctimeout() 함수 안에서 tpreturn()하는 경우에 소스 코드에 정의된 인자값을 무시하고 환경설정에 정의된 값을 강제로 설정한다.

TPRETURN 옵션의 설정 가능한 값은 다음과 같다.

항목	설명
FAIL	TPFAIL 인자를 강제로 설정한다.
EXIT	TPEXIT 인자를 강제로 설정한다.
DOWN	TPDOWN 인자를 강제로 설정한다.
NONE	소스코드에 정의된 값을 그대로 사용한다.

- 관련 이슈 : [IMS 171100](#)

• **Tmax Failover Retry 기능 추가**

- TMMOPT 절에 -R count로 retry할 횟수를 지정한다. 이 횟수 \* nliveinq 시간이 경과한 후에 failover를 수행한다.
- TMMOPT 절에 -D로 지정할 경우 failback될 때 자동으로 기동된 backup svr을 종료한다.

- 관련 이슈 : [IMS 105883](#)

• **TMMOPT 절에 -R count, -D 옵션 동적 설정 기능 추가**

- TMMOPT 절에 -R count, -D 옵션을 tmdm -m setopt로 동적 설정이 가능하도록 기능을 추가하였다.

```
$$1 tmaxi4 (tadm): setopt -tmm -R 5
new value (5) is set for section = -tmm, name = N/A, fld = -R
(I) TMM3039 TMM option [-R] set to 5 [TMM0295]

$$1 tmaxi4 (tadm): setopt -tmm -D 0
new value (0) is set for section = -tmm, name = N/A, fld = -D
(I) TMM3039 TMM option [-D] set to 0 [TMM0295]
```

- 관련 이슈 : [IMS 112688](#)



## • tmdown을 실행할 때 처리 중인 거래를 보장하는 기능 추가

- GATEWAY(TMAX, TMAXNONTX) 절 RDOWN=Y 설정
  - 도메인 게이트웨이로 인/아웃바운드를 요청 이후에 tmdown을 수행할 때 수행 중인 거래가 종료된 후 down을 수행한다.
  - tmdown 중 게이트웨이로 들어온 요청에 대해서 바로 에러 리턴(TPENOREADY)을 수행한다.
  - tmdown 중 타임아웃이 발생하면 에러 리턴(TPETIME) 후 down을 수행한다.
- NODE 절 CLIREQBLOCK=Y 설정
  - 클라이언트 요청 이후에 tmdown을 수행할 때 수행 중인 거래가 종료된 후 down을 수행한다.
  - tmdown 중 클라이언트 호출에 대해서 바로 에러 리턴(TPENOREADY)을 수행한다.
  - tmdadmin -m 명령어를 실행한 뒤 set -n nodename crb Y/N으로 동적 변경 가능하다.
- 관련 이슈 : [IMS 108322](#)

## • TSM 엔진 추가

- 기존에 tmapm 프로그램을 별도로 제공하고 있으나, 이를 사용하기 위해서는 사용자가 환경설정 파일에 UCS 서버로 직접 등록해야 하고, 이 프로그램에서 타임아웃을 감지할 때에 사용자가 정의한 스크립트를 호출하고 끝나기 때문에 사용하기 번거로운 부분이 존재한다. 따라서 tmapm 프로그램을 기존에 비해 설정하기 편한 엔진 프로세스(TSM)로 관리할 수 있도록 하였다.

### ◦ 환경설정

- SERVER 절 TMAPM=n (기본값: -1, 설정 범위: 1~32767)

TSM 엔진 프로세스가 해당 서버에서 수행되는 서비스에 대한 타임아웃을 체크하도록 설정한다. -1을 설정하면 해당 서버는 TSM의 체크 대상에서 제외된다. 설정값은 타임아웃 발생 이후 체류 가능한 시간을 의미한다. 만약 1을 설정하면 타임아웃이 발생한 이후에 1초동안 프로그램이 수행될 시간을 제공한 뒤에 해당 프로세스를 강제종료(SIGKILL)시킨다. 최대 32767초까지 설정할 수 있다.

만약 모든 SERVER가 TMAPM을 설정하지 않을 경우에는 TSM 엔진 프로세스는 tmbboot 시점에 기동하지 않는다. 그러나 동적 설정 변경으로 SERVER에 대한 TMAPM을 설정하면 tsm 엔진 프로세스가 즉시 기동하게 된다. tmbboot -e tsm 명령어로 기동할 수 있다.

- SERVICE 절 TMAPM=[Y|n] (기본값: Y)

TSM 엔진 프로세스가 해당 서비스에 대해서 타임아웃을 체크할 것인지 여부를 설정한다. 만약 특정 서비스를 N으로 설정하면, 해당 서비스에서 타임아웃이 발생하더라도 TSM에서 체크하지 않는다. 이 옵션은 tmapm 프로그램에서도 동일하게 적용된다. (기존부터 제공되던 옵션)

### ◦ 동적 설정 방법

- set -v svr\_name tmapm n 명령을 통해 동적으로 변경이 가능하다.

```
$$1 tmax (tmdm): echo set -v alarm0 ta 5
new value (5) is set for section = SERVER, name = alarm0, fld = ta
$$2 tmax (tmdm): ADM quit for node (tmax)
```

### ◦ 발생 로그

- (E) BOOT3025 tsm is already running [BOOT0040]

tmboot -e tsm 명령을 수행할 때 이미 기동 중인 경우 발생하는 에러이다.

- (E) CFL3198 line n : invalid parameter value : TMAPM. Valid range = (MIN : -1, MAX : 32767) except (0) [CFL5542]

SERVER 절 TMAPM 옵션을 설정할 때 잘못 설정한 경우 발생하는 에러이다.

- (W) TSM7203 svc(service\_name) of svr(svr\_name:spri:pid) clh(n) svctime(n) elapsed n seconds [TSM0304]

TSM에서 체크한 서버 프로세스의 서비스 타임아웃 정보를 출력한다.

- (I) TSM7204 SVR (svr\_name) spri(n) is down due to TMAPM=n [TSM0306]

TSM에서 해당 서버 프로세스를 SIGKILL 시그널로 강제종료했음을 알리는 로그이다.

- (I) TMM3068 TSM (tsm) is started, pid = n[TMM5205]

동적 설정 변경으로 인해 TSM 프로세스가 새롭게 기동하였음을 알리는 로그이다.

◦ 유의사항

- tpsvctimeout()에서 tpreturn(TPEXIT)에 의한 서버 재기동은 서버 프로세스 자신이 직접 재기동을 수행한다. 반면에 TSM에 의한 타임아웃은 SIGKILL 시그널에 의해 서버 프로세스가 강제 종료되는 것이다. 따라서 클라이언트가 리턴받는 에러 코드가 TPESVCERR가 아닌 TPESVRDOWN이 된다.
- multi-clh 환경에서 해당 서버에 동시에 스케줄링되는 상황에서 먼저 수행되는 서비스가 타임아웃 발생하는 경우, 서버 자신이 직접 재기동하는 경우에는 아직 대기 중인 다른 CLH의 요청이 다른 프로세스에서 정상적으로 처리되는 반면에, TSM에 의해서 강제 종료되는 경우 대기 중이던 다른 CLH의 요청들도 TPESVRDOWN 에러를 받게 된다.

◦ 관련 이슈 : [IMS 110953](#), [IMS 110305](#)

• 다른 노드로 스케줄링 된 요청에 대해서 상대노드 장애 시 에러 응답 받는 기능 추가

- DOMAIN 절, NODE 절에 CLHRRCHK=Y로 설정하는 경우 CLH에서 tpcall, tpacall, tprecvfromcli 요청에 대해서 다른 노드로 스케줄링될 경우 해당 노드의 장애를 감지하여 에러(TPESVRDOWN)를 주도록 추가하였다.
- 관련 이슈 : [IMS 193880](#)

• broadcast에 대한 타임아웃 기능 추가

- CLHOPT 절에 [-B] 옵션 추가

항목	설명
[ -B sec ]	메시지 삭제 시점을 설정한다. BROADCASTTIMEOUT 지난 메시지가 있을 때 새로운 broadcast enq 요청이 들어 오면 이때 큐를 체크하여 삭제한다.

- 패치 모듈 : CLH
- 관련 이슈 : [IMS 102184](#)

• **cfgopt tmm에 -R, -D 추가 및 -gw 기능추가**

- cfgopt -tmm을 조회할 때 -R, -D 옵션 항목을 추가하였다.
- cfgopt -gw gwname 명령어를 추가하였다.
- 예제

```

$ tadmin
$$1 node1 (tmadm): setopt
Usage: setopt -tmm -option value
        -gw gwname -option value

$$2 node1 (tmadm): cfgopt
Usage: configopt -tmm
        -gw gwname

$$3 node1 (tmadm): setopt -tmm -D 1
new value (1) is set for section = -tmm, name = N/A, fld = -D

$$4 node1 (tmadm): cfgopt -tmm
-----
-tmm configurable value
-----
accept retry count          (-A) = 100
max forked threshold        (-F) = 95
booting timeout             (-t) = 10
auto failback               (-D) = 1
node check try count        (-R) = 0
-----

$$5 node1 (tmadm): setopt -gw gw_name -D 2
new value (2) is set for section = -gw, name = gw_name, fld = -D

$$6 node1 (tmadm): cfgopt -gw gw_name
-----
-gw configurable value
-----
debug log level            (-D) = 2
-----

```

- 관련 이슈 : [IMS 118505](#)

## 2.1.2. 클라이언트/서버

### • 구 버전 클라이언트 에러 코드 지원 옵션 추가

- 환경설정 파일(tmax.env)에 TMAX\_PRB\_1000=Y를 지정하는 경우 Tmax 6 버전과 Tmax 5 SP2 Fix#2 버전 이전의 동작 방식으로 강제 롤백한다.
- 채널 끊긴 후에 호출하는 경우 TPEOS가 리턴하는 문제를 TPECLOSE가 리턴하도록 수정한 부분을 기존대로 TPEOS가 발생하도록 동작한다.
- CLH와 클라이언트의 채널 연결 상태에 문제가 있어서 에러가 발생한 경우(예를 들어 tpcall(), tmax\_chk\_conn() 등의 API에서 TPECLOSE, TPESYSTEM 에러가 발생한 경우), tpend() 호출하지 않고 클라이언트 API를 호출하는 경우 TPECLOSE 에러가 발생하도록 수정한 부분을 기존대로 TPEOS가 발생하도록 동작한다.
- 관련 이슈 : [IMS 111005](#)

### • tpgetsvglst\_bynode(), tpgetnodelist() API 추가

- 특정 노드에 존재하는 서비스 그룹으로 tpcallsvg() 요청을 할 때 해당 API의 svgno 인자에 넣을 값을 검색하기 편리하게 하기 위해 제공한다.
- 관련 이슈 : [IMS 144454](#)

#### [tpgetsvglst\_bynode()]

지정한 서비스를 제공하는 서버 그룹 중에 지정한 노드명에 해당하는 서버 그룹 목록을 가져온다. tpgetsvglst()와 동일하지만, 노드명 인자에 따라 서버 그룹 목록의 내용이 달라진다.

- 프로토타입

```
struct svglst *tpgetsvglst_bynode(char *nodename, char *svc, long flags);
```

- 파라미터

파라미터	설명
nodename	svc에 지정한 서비스를 제공하는 노드명을 입력한다.  NULL 을 입력할 경우 tpgetsvglst()와 동일한 동작을 수행한다.  만약 잘못된 노드명을 입력하는 경우 TPEINVAL 또는 TPENOENT 오류가 발생한다.
svc	서비스명을 입력한다.
flags	현재는 사용되지 않는다.

- 반환값

에러가 발생하면 NULL을 반환하고 tperrno에 상황에 해당하는 값이 설정된다. 그렇지 않으면 서비스 호출에 실패한 서버 그룹 리스트를 struct svglst에 설정하여 반환한다.

반환값	설명
NULL이 아닌 값	함수 호출에 성공한 경우이다. 지정한 노드명에 해당하는 서버 그룹 목록이 리턴된다.
NULL	함수 호출에 실패한 경우이다. tperrno에 에러 코드가 설정된다.

◦ 예제

```
#include <stdio.h>
#include <usrinc/atmi.h>

SVGCALL(TPSVCINFO *msg)
{
    char *svcname = msg->data;

    struct nodelist *node = tpgetnodelist();
    if (node == NULL)
        ...error process

    for (i = 0; i < node->count; i++) {
        printf("[%d] %16s : %x:%p\n", i, node->nodelist[i].nodename,
            node->nodelist[i].ipaddr, node->nodelist[i].port);
    }

    ...

    struct svglist *svg = tpgetsvglist_bynode(node->nodelist[1].nodename, svcname, 0);
    if (svg == NULL)
        ...error process

    for (i = 0; i < svg->ns_entry; i++) {
        printf("[%d] %d(%x)\n", i, svg->s_list[i], svg->s_list[i]);
    }

    ...

    tpreturn(TPSUCCESS,0,(char *)msg->data, msg->len,0);
}

```

**[tpgetnodelist()]**

현재 접속된 Tmax 시스템의 노드의 목록을 가져오는 함수이다. 이때 반환된 nodelist 구조체의 주소는 메모리 해제를 수행해서는 안된다.

◦ 프로토타입

```
struct nodelist *tpgetnodelist()
```

◦ 반환값

에러가 발생하면 NULL을 반환하고 tperrno에 상황에 해당하는 값이 설정된다. 그렇지 않으면 서비스 호출에 실패한 서버 그룹 리스트를 struct svglist에 설정하여 반환한다.

반환값	설명
NULL이 아닌 값	함수 호출에 성공한 경우이다. 노드들의 리스트가 담긴 구조체 리스트를 리턴한다.
NULL	함수 호출에 실패한 경우이다. tperrno에 에러 코드가 설정된다.

◦ 예제

```

main(int argc, char *argv[])
{
    int    ret;
    int    starti = 0, endi = 0;

    if (argc != 2) {
        printf("Usage: argv[1] string\n");
        exit(1);
        if((ret = tmaxreadenv( "tmax.env", "TMAX" )) == -1 ){
            printf("<%-15s> tmaxreadenv fail [%s]", __FILE__, tpstrerror(tperrno));
            exit(1);
        }

        if (tpstart((TPSTART_T *)NULL) == -1){
            printf("<%-15s> tpstart fail [%s]", __FILE__, tpstrerror(tperrno));
            exit(1);
        }

        /* tpgetsprlist in client */
        ret = tpgetsprlist(argv[1], 2, &starti, &endi, 0);
        if (ret < 0) {
            printf("<%-15s> tpgetsprlist fail [%s]", __FILE__, tpstrerror(tperrno));
            exit(1);
        }
        else {
            printf("Received Message : starti[%d], endi[%d]\n", starti, endi);
        }
        tpend();
    }
}

```

• CliWatcherCallback(), tpsetcliwatcher(), tpclrcliwatcher() API 추가

- 클라이언트가 종료되었을 때 UCS 서버에서 이를 감지하고 후속 작업을 할 수 있는 API를 제공한다.
- 관련 이슈 : [IMS 146611](#)

**[CliWatcherCallback()]**

클라이언트 종료를 통지받을 콜백함수이며, 사용자가 함수를 작성한 뒤 tpsetcliwatcher 함수의 callback 인자에 넣어준다. 콜백함수가 호출되는 시점은 tpschedule()이 호출되고 클라이언트의 종료 이벤트가 감지되는 시점에 호출된다.

- 프로토타입

```

typedef void (*CliWatcherCallback)(int clid, int reason, void *args);

```

- 파라미터

파라미터	설명
clid	종료된 클라이언트의 clid이다.
reason	클라이언트가 종료된 원인을 명시한다.  tmaxgw_setcliwatcher()의 경우 요청을 했으나 clh에서 감지 못하는 경우에도 이 콜백함수를 통해 응답을 준다.
args	tpsetcliwatcher()나 tmaxgw_setcliwatcher()에서 사용자가 정의한 args 변수의 값이다. 사용자 정의 데이터를 전달하는 용도로 사용된다. 해당 args를 사용할 때는 반드시 메모리 할당/해제에 주의해야 한다.

- 예제

```
#include <stdio.h>
#include <usrinc/atmi.h>
#include <usrinc/ucs.h>

void logout_callback(int clid, int reason, void *args) {
    printf("[%s] clid:%#x, args:%p, reason:%d\n", "CALLBACK", clid, args, reason);
}

LOGIN(TPSVCINFO *msg) {
    int clid, ret;
    time_t curtime = time(NULL);

    clid = tpgetclid();
    ret = tpsetcliwatcher(clid, logout_callback, (void *)curtime, TPNOFLAGS);

    printf("[%s] clid:%#x, curtime:%ld, set_watch:%d(%s), data:%s\n",
           msg->name, clid, curtime, ret, tpstrerror(tperrno), msg->data);

    tpreturn(TPSUCCESS, 0, NULL, 0, 0);
}
```

### [tpsetcliwatcher()]

clid에 대응하는 클라이언트가 종료될 때 콜백함수가 호출되도록 설정한다. 클라이언트가 접속한 clh에 등록하는 과정이 끝나야 리턴된다. 콜백함수가 호출되는 시점은 tpschedule()이 호출되고 클라이언트의 종료 이벤트가 감지되는 시점이다.

- 프로토타입

```
int tpsetcliwatcher(int clid, CliWatcherCallback callback, void *args, int flags);
```

- 파라미터

파라미터	설명
clid	tpgetclid() 등으로 확인한 클라이언트의 id를 입력한다. 클라이언트가 아닌 서버 등의 clid를 입력하면 TPEINVAL 에러가 발생한다.
callback	클라이언트가 종료될 때 통지받을 함수를 위의 CliWatcherCallback 타입으로 생성하여 입력한다.
args	클라이언트가 종료될 때 콜백함수가 호출 될 때 인자로 받을 변수를 입력한다. 사용자가 자유롭게 정의할 수 있다.
flags	현재 사용되지 않는다.

◦ 반환값

반환값	설명
1	함수 호출에 성공한 경우이다.
-1	함수 호출에 실패한 경우이다. tperrno에 에러 코드가 설정된다.

◦ 예제

```
#include <stdio.h>
#include <usrinc/atmi.h>
#include <usrinc/ucs.h>

void logout_callback(int clid, int reason, void *args) {
    printf("[%s] clid:%#x, args:%p, reason:%d\n", "CALLBACK", clid, args, reason);
}

LOGIN(TPSVCINFO *msg) {
    int clid, ret;
    time_t curtime = time(NULL);

    clid = tpgetclid();
    ret = tpsetcliwatcher(clid, logout_callback, (void *)curtime, TPNOFLAGS);

    printf("[%s] clid:%#x, curtime:%ld, set_watch:%d(%s), data:%s\n",
           msg->name, clid, curtime, ret, tpstrerror(tperrno), msg->data);

    tpreturn(TPSUCCESS, 0, NULL, 0, 0);
}
```

**[tpclrcliwatcher()]**

tpsetcliwatcher()으로 등록한 clid에 대해서 종료 이벤트 감지를 취소한다. tpclrcliwatcher() 함수가 호출되고 나면 클라이언트가 종료되었을 때 더이상 콜백은 호출되지 않는다.

◦ 프로토타입

```
int tpclrcliwatcher(int clid, void **pargs, int flags);
```



- 파라미터

파라미터	설명
clid	종료 이벤트 감지를 취소할 clid를 입력한다.
pargs	tpsetcliwatcher()에서 입력한 args 변수의 값을 되돌려준다.
flags	현재 사용되지 않는다.

- 반환값

반환값	설명
1	함수 호출에 성공한 경우이다.
-1	함수 호출에 실패한 경우이다. tperrno에 에러 코드가 설정된다.

- 예제

```
#include <stdio.h>
#include <usrinc/atmi.h>
#include <usrinc/ucs.h>

LOGOUT(TPVCINFO *msg) {
    int clid, ret;
    void *args;
    time_t curtime;

    clid = tpgetclid();
    ret = tpclrcliwatcher(clid, &args, TPNOFLAGS);

    printf("[%s] clid:##x, args:%p, clr_watch:%d(%s), data:%s\n",
           msg->name, clid, args, ret, tpstrerror(tperrno), msg->data);

    tpreturn(TPSUCCESS, 0, NULL, 0, 0);
}
```

## • CliWatcherCallback(), tpsetcliwatcher(), tpclrcliwatcher() API 추가

- 클라이언트가 종료되었을 때 CustomGateway 서버에서 이를 감지하고 후속 작업을 할 수 있는 API를 제공한다.
- 관련 이슈 : [IMS 146611](#)

### [CliWatcherCallback()]

클라이언트 종료를 통지받을 콜백함수이며, 사용자가 함수를 작성한 뒤 tpsetcliwatcher 함수의 callback 인자에 넣어준다. 콜백함수가 호출되는 시점은 tpschedule()이 호출되고 클라이언트의 종료 이벤트가 감지되는 시점에 호출된다.

- 프로토타입

```
typedef void (*CliWatcherCallback)(int clid, int reason, void *args);
```

- 파라미터

파라미터	설명
clid	종료된 클라이언트의 clid이다.
reason	클라이언트가 종료된 원인을 명시한다.  tmaxgw_setcliwatcher()의 경우 요청을 했으나 clh에서 감지 못하는 경우에도 이 콜백함수를 통해 응답을 준다.
args	tpsetcliwatcher()나 tmaxgw_setcliwatcher()에서 사용자가 정의한 args 변수의 값이다. 사용자 정의 데이터를 전달하는 용도로 사용된다. 해당 args를 사용할 때는 반드시 메모리 할당/해제에 주의해야 한다.

- 예제

```
#include <stdio.h>
#include <usrinc/atmi.h>
#include <usrinc/ucs.h>

void logout_callback(int clid, int reason, void *args) {
    printf("[%s] clid:%#x, args:%p, reason:%d\n", "CALLBACK", clid, args, reason);
}

LOGIN(TPSVCINFO *msg) {
    int clid, ret;
    time_t curtime = time(NULL);

    clid = tpgetclid();
    ret = tpsetcliwatcher(clid, logout_callback, (void *)curtime, TPNOFLAGS);

    printf("[%s] clid:%#x, curtime:%ld, set_watch:%d(%s), data:%s\n",
           msg->name, clid, curtime, ret, tpstrerror(tperrno), msg->data);

    tpreturn(TPSUCCESS, 0, NULL, 0, 0);
}
```

### [tpsetcliwatcher()]

clid에 대응하는 클라이언트가 종료될 때 콜백함수가 호출되도록 설정한다. 클라이언트가 접속한 clh에 등록하는 과정이 끝나야 리턴된다. 콜백함수가 호출되는 시점은 tpschedule()이 호출되고 클라이언트의 종료 이벤트가 감지되는 시점이다.

- 프로토타입

```
int tpsetcliwatcher(int clid, CliWatcherCallback callback, void *args, int flags);
```

- 파라미터

파라미터	설명
clid	tpgetclid() 등으로 확인한 클라이언트의 id를 입력한다. 클라이언트가 아닌 서버 등의 clid를 입력하면 TPEINVAL 에러가 발생한다.
callback	클라이언트가 종료될 때 통지받을 함수를 위의 CliWatcherCallback 타입으로 생성하여 입력한다.
args	클라이언트가 종료될 때 콜백함수가 호출 될 때 인자로 받을 변수를 입력한다. 사용자가 자유롭게 정의할 수 있다.
flags	현재 사용되지 않는다.

◦ 반환값

반환값	설명
1	함수 호출에 성공한 경우이다.
-1	함수 호출에 실패한 경우이다. tperrno에 에러 코드가 설정된다.

◦ 예제

```
#include <stdio.h>
#include <usrinc/atmi.h>
#include <usrinc/ucs.h>

void logout_callback(int clid, int reason, void *args) {
    printf("[%s] clid:%#x, args:%p, reason:%d\n", "CALLBACK", clid, args, reason);
}

LOGIN(TPSVCINFO *msg) {
    int clid, ret;
    time_t curtime = time(NULL);

    clid = tpgetclid();
    ret = tpsetcliwatcher(clid, logout_callback, (void *)curtime, TPNOFLAGS);

    printf("[%s] clid:%#x, curtime:%ld, set_watch:%d(%s), data:%s\n",
           msg->name, clid, curtime, ret, tpstrerror(tperrno), msg->data);

    tpreturn(TPSUCCESS, 0, NULL, 0, 0);
}
```

**[tpclrcliwatcher()]**

tpsetcliwatcher()으로 등록한 clid에 대해서 종료 이벤트 감지를 취소한다. tpclrcliwatcher() 함수가 호출되고 나면 클라이언트가 종료되었을 때 더이상 콜백은 호출되지 않는다.

◦ 프로토타입

```
int tpclrcliwatcher(int clid, void **pargs, int flags);
```

◦ 파라미터

파라미터	설명
clid	종료 이벤트 감지를 취소할 clid를 입력한다.
pargs	tpsetcliwatcher()에서 입력한 args 변수의 값을 되돌려준다.
flags	현재 사용되지 않는다.

◦ 반환값

반환값	설명
1	함수 호출에 성공한 경우이다.
-1	함수 호출에 실패한 경우이다. tperrno에 에러 코드가 설정된다.

◦ 예제

```
#include <stdio.h>
#include <usrinc/atmi.h>
#include <usrinc/ucs.h>

LOGOUT(TPVCINFO *msg) {
    int clid, ret;
    void *args;
    time_t curtime;

    clid = tpgetclid();
    ret = tpclrcliwatcher(clid, &args, TPNOFLAGS);

    printf("[%s] clid:%#x, args:%p, clr_watch:%d(%s), data:%s\n",
           msg->name, clid, args, ret, tpstrerror(tperrno), msg->data);

    tpreturn(TPSUCCESS, 0, NULL, 0, 0);
}
```

• MQ xa\_stub 라이브러리 추가

- libmq.so 라이브러리를 추가하였다.
- MultipleRM에서 SVRGROUP 절 DBNAME=MQ로 설정하면 해당 stub 라이브러리를 사용할 수 있다.
- 관련 이슈 : [IMS 107602](#)

• MultipleRM stub 라이브러리를 지정할 수 있는 기능 추가

- \$TMAXDIR/config/RM 파일에 Tmax 환경설정의 DBNAME과 일치하는 항목이 있으면 해당 라이브러리를 사용할 수 있도록 기능을 추가하였다.

```
*SVRGROUP
svg1          NODENAME = "@HOSTNAME@",
              OPENINFO = "QMNAME=MQTEST",
              TMSNAME  = tms_mq,
DBNAME = MQ_RENAME
```

```
[RM file]
# Sample for RM Definition
# Oracle
Oracle_XA:oras:-L${ORACLE_HOME}/lib -lcIntsh `cat ${ORACLE_HOME}/lib/ldflags` `cat
${ORACLE_HOME}/lib/sysliblist`
# Informix
Informix_XA:infs:-L${INFORMIXDIR}/lib -L${INFORMIXDIR}/lib/esql -lifxa `esql -libs`
# Sybase
Sybase_XA:sybs:-L${SYBASE}/${SYBASE_OCS}/lib -lxatm `sybopts.sh ct cs comlibs` `sybopts.sh
syslibs`
# DB2
DB2_XA:db2s:-L${DB2DIR}/lib -ldb2
MQ_RENAME:mqs
```

- 관련 이슈 : [IMS 107602](#)

- **tmax\_my\_rminfo() API 추가**

- MultipleRM 서버에서 자신이 속한 서버 그룹의 DB 정보를 가져오는 API를 추가하였다.

- 관련 이슈 : [IMS 113300](#)

### [tmax\_my\_rminfo()]

- 프로토타입

```
int tmax_my_rminfo(RM_INFO_T *rminfo, int max_elem);
```

- 파라미터

파라미터	설명
rminfo	M 정보를 저장할 구조체 배열 버퍼의 포인터이다.
max_elem	구조체 배열의 요소 개수이다(MRM은 서버 그룹을 최대 16개까지 지정 가능하다).

다음은 RM\_INFO\_T 구조체 정보이다.

```
#include <usrinc/tmaxapi.h>

typedef struct RM_INFO_S {
    int rmid;
    int flags;
    char dbname[DBNAME_SIZE + 1];
    char openinfo[RM_STRING_SIZE + 1];
    char closeinfo[RM_STRING_SIZE + 1];
} RM_INFO_T;
```

- 반환값

반환값	설명
RM 개수	자기 자신이 속한 서버 그룹이 가지는 RM의 개수

### • tpgetcliinfo() API 추가

- Tmax 시스템에 접속된 클라이언트 중 clid에 해당하는 클라이언트의 정보를 얻어 오는 tpgetcliinfo() 함수를 추가 하였다.
- 관련 이슈 : [IMS 129894](#)

### [tpgetcliinfo()]

- 프로토타입

```
#include <tmadmin.h>
int tpgetcliinfo(int clid, struct tmadm_cliinfo_body *info)
```

- 파라미터

파라미터	설명
clid	조회할 clid이다.
info	clid와 일치하는 클라이언트의 정보가 담기는 곳이다.

- 반환값

반환값	설명
1	함수 호출에 성공한 경우이다.
-1	함수 호출에 실패한 경우이다. tperrno에 에러 코드가 설정된다.

반환한 struct tmadm\_cliinfo\_body의 내용은 다음과 같다.

```
struct tmadm_cliinfo_body {
    int no; /* cli index */
    int clid; /* CLID */
    int clhno;
    int count;
    int idle;
    int reserve_int[3];
    char status[TMAX_NAME_SIZE];
    char addr[TMAX_IPADDR_SIZE];
    char username[TMAX_NAME_SIZE];
    int qpcount;
    int emaxcount;
    int reserved1;
    int reserved2;
};
```

## • libsvrucsmnt.a 라이브러리 추가

- UCS의 usermain에서 처리하는 비즈니스 로직을 MultiThread로 동작하도록 libsvrucsmnt.a 라이브러리를 추가하였다.
- 관련 이슈 : [IMS 140461](#)

## • XA 서버 그룹의 서버에서 DB Stub library를 링크하지 않아도 XA 처리하는 기능 추가

- DB를 설정한 XA 서버 그룹에 속하는 서버가 해당 DB의 xa stub 및 db client library를 링크하지 않는 경우에 라이브러리 로딩이 실패하여 xa\_switch\_t 심볼을 찾지 못하는 문제를 해결하기 위해 서버 프로그램을 변경하지 않고 사용자가 명시한 라이브러리들을 모두 로드하는 기능을 추가하였다.
- 서버 그룹 TYPE이 MTMAX(MultipleRM 기능) 타입, STMAX, TMAX 타입 모두 아래의 동작 순서에 따라 xa\_switch\_t 심볼을 탐색한다.
  1. 자기 자신의 프로그램 이미지에 \_tmax\_xasw\_init() 함수가 존재하는지 확인한다.
  2. RM 설정파일에 SVRGROUP 절의 DBNAME 항목과 동일한 이름의 설정에서 라이브러리 목록을 추출한다.
  3. RM 파일이 없거나 SVRGROUP 절의 DBNAME 항목과 동일한 설정이 없는 경우 내부적으로 정의된 라이브러리 목록을 추출한다.
- \$TMAXDIR/config/RM 설정 파일의 정의 방법

RM 파일에 Tmax 환경설정의 DBNAME과 일치하는 항목이 있으면 해당 library를 동적으로 로딩하여 사용한다.

- 기본 형식 : 각 라인마다 DBNAME:library[,library] 형식으로 작성한다.
- DBNAME : SVRGROUP 절의 DBNAME 항목에 정의한 이름과 동일한 이름으로 설정한다.
- library : 로드할 라이브러리 이름을 명시한다. 하나 이상의 라이브러리를 로드할 경우 콤마(,)를 통해 여러 개를 정의한다.

- RM 파일 예제)

```
ORACLE:oras
ORACLE_ALL:c\ntsh,oras
ORACLE_FULL:libc\ntsh.so,liboras.so
TIBERO:tbs
TIBERO_ALL:tbxa,tbs
TIBERO_FULL:libtbxa.so,libtbs.so
```

- 관련 이슈 : [IMS 160532](#)

## • tprecvfromcli() API 추가

- OpenFrame 전용 함수로 서버에서 클라이언트로 메시지를 요청하고 받는 함수를 추가하였다.
- 이 기능은 tpcall과 같이 블로킹으로 동작하며, clid는 tpgetclid로 얻어온 clid를 사용한다. 다른 인자는 tpcall과 동일하게 사용된다. 반환값은 실패하는 경우 -1, 성공하는 경우 ATMI\_RETURN을 리턴한다.
- 프로토타입

```
#include <tmadmin.h>
int __EXPORT tprecvfromcli(int clid, char* idata, long ilenl, char **odata, long *olen, long
```

```
flags1);
```

- 관련 이슈 : [IMS 94632](#)

#### • CICS의 SYNCPOINT와 같은 기능의 트랜잭션 처리 기능을 추가

- OpenFrame 전용 함수로 트랜잭션이 하나 이상의 서비스에서 수행중인 상태에서 모든 서비스들이 tpsync()를 호출하면 호출된 서비스의 계층 구조에서 마지막에 호출된 서비스부터 xa\_end()를 호출한 뒤에 트랜잭션 개시자에게로 tpsync()의 요청이 전달된다. 최종적으로 트랜잭션 내의 모든 서비스들이 tpsync()를 호출했을 때 트랜잭션 개시자가 tx\_commit(), tx\_begin()을 수행하고, 새롭게 시작된 트랜잭션의 정보를 다시 호출된 서비스들에게 전달해서 모두 같은 트랜잭션 정보를 가지고 xa\_start를 수행하도록 한다.
- tpsync(), tpsync\_rollback(), tpsync\_set\_timeout(), tpsync\_get\_timeout() API 를 제공한다.
- 프로토타입

```
#include <tmadmin.h>
int __EXPORT tpsync(int *ptx_return, long flags);
int __EXPORT tpsync_rollback(int *ptx_return, long flags);
int __EXPORT tpsync_set_timeout(int blocktime);
int __EXPORT tpsync_get_timeout();
```

- 관련 이슈 : [IMS 94632](#)

#### • tpsendto(), tprecvfrom() API 추가

- OpenFrame 전용 함수로 비요청 통신용 tpsendto()와 tprecvfrom() 서버/클라이언트 함수를 추가하였다.
- 클라이언트로부터 tpacall 요청을 받은 서버가 최초로 호출할 수 있다.
- 프로토타입

```
#include <tmadmin.h>
int __EXPORT tpsendto(int clid, char *data, long len, long flags);
int __EXPORT tprecvfrom(TPMSGDESC *desc, char **data, long *len, long flags);
```

- 관련 이슈 : [IMS 94632](#)

## 2.1.3. 유틸리티

#### • tmboot -W 옵션 추가

- tmboot를 수행하는 경우 현재 기동 중인 서버 프로세스가 완전히 기동이 될 때까지 다음 프로세스를 기동시키지 않고 대기하는 기능을 추가했다.

#### • Tmax 버전 정보 patch/release 여부 정보 기능 추가

- Tmax 버전 정보를 출력할 때 릴리즈 버전인 경우(final\_release), 패치 버전인 경우(patch)로 표시한다.

```
$ tmaxlibver -l libsvr.so -d -6
libsvr.so for TMAX Version 5.0 SP #2 Fix #3 r10353 fd(16384) 64bit binary for Solaris 11
```



```
(patch)
$ tadmin -V
Admin for TMAX Version 5.0 SP #2 Fix #3 r10353 fd(16384) 64bit binary for Solaris 11 (patch)
$ svr2 -V
SVR for TMAX Version 5.0 SP #2 Fix #3 r10353 fd(16384) 64bit binary for Solaris 11 (patch)
```

◦ 관련 이슈 : [IMS 107212](#)

• **cfl 공유 메모리 크기 출력 -P 옵션 추가**

◦ cfl을 실행할 때 Tmax에서 사용하는 전체 공유 메모리의 크기를 보여주는 -P 옵션을 추가한다.

```
$ cfl -i m.m -P
shmsize info:
shmkey = 0x11936(71990), shmsize = 552652
shmkey = 0x11937(71991), shmsize = 290192
shmkey = 0x11938(71992), shmsize = 230816
shmkey = 0x11939(71993), shmsize = 49152
shmkey = 0x1193a(71994), shmsize = 0
CFL is done successfully for node
```

◦ 관련 이슈 : [IMS 108588](#)

• **svcrpt -A 옵션 추가**

◦ svcrpt에서 서비스 로그를 정렬할 때 평균 처리시간으로 정렬하는 -A 옵션을 추가한다.

```
$ svcrpt -i /home/tmaxha/tmax/log/ulog/svclog* -A

Services issued
    between Thu May 26 18:34:31 2016
    and      Thu May 26 18:35:04 2016
```

svc_name	18 - 19 num/avg	total num/avg
SVC10	1000/1.000	1000/1.000
SVC9	1000/0.900	1000/0.900
SVC8	1000/0.800	1000/0.800
SVC7	1000/0.700	1000/0.700
SVC6	1000/0.600	1000/0.600
SVC5	1000/0.500	1000/0.500
SVC4	2000/0.400	2000/0.400
SVC3	1000/0.300	1000/0.300
SVC2	2000/0.200	2000/0.200
SVC1	4000/0.100	4000/0.100
total	15000/0.427	15000/0.427

◦ 관련 이슈 : [IMS 111395](#)

• **12개월 데모 라이선스 발급 추가**

◦ 4개월(demo, D 옵션), 6개월(long demo, L 옵션), 12개월(year demo, Y 옵션)로 데모 라이선스 종류 분리 및 적용하였다.

- **cfl -g** 옵션 추가

- 게이트웨이에서 REMOTE IP/PORT에 대해 중복 설정이 가능하도록 옵션이 추가되었다.
- 예제

```
cfl -g -i gw.m  
CFL is done successfully for node(tmaxi4)
```

- 관련 이슈 : [IMS 117102](#)

## 2.1.4. 관리도구

- **reconnect\_clh** 기능 추가

- CLH와 다른 노드의 재연결을 시도할 수 있다. 연결 상태에 따라 끊어진 연결만 시도하는 경우와 모든 연결을 재연결하는 방식 두 가지 모두 가능하다.
- 사용 방법

```
$$1 tmax1 (tmadm): rc nodename [-a]
```

항목	설명
[-a]	모든 연결을 재연결 한다.

- 관련 이슈 : [IMS 96544](#)

- **clientinfo -c, -P, -p** 옵션 추가

- clientinfo를 수행할 때 CRYPTPORT, 특정 포트에 접속한 클라이언트만 조회할 수 있는 기능을 추가하였다.
- 사용 방법

```
$$1 tmax1 (tmadm): ci [-c] [-P] [-p [portno, ...]]
```

옵션	설명
[-c]	암호화 모드로 접속한 클라이언트만 조회할 수 있다.
[-P]	포트 번호 정보를 추가로 출력한다.
[-p [portno, ...]]	열거한 Tmax 접속 포트 번호에 대해서만 출력한다.

- 예제

다음은 명령어 사용 예제이다.

```
$$1 tmaxi9 (tmadm): ci -c
```

CLH 0:

```
-----  
-----  
cli_id   clid   status count  lastin_time  ipaddr  
usrname          tmax_port  
-----  
-----  
      0 0x00000000 RUN      0          1 127.0.0.1:58235  
7189  
      1 0x00000001 RUN      0          1 127.0.0.1:58236  
7389  
      4 0x00000004 RUN      0          1 127.0.0.1:58239  
7289  
-----  
-----
```

Total Connected Clients(CRYPTED) = 3

\$\$\$ tmaxi9 (tmadm): ci -p 7189,7289,7389,7489,7589

CLH 0:

```
-----  
-----  
cli_id   clid   status count  lastin_time  ipaddr  
usrname          tmax_port  
-----  
-----  
      0 0x00000000 RUN      0          1 127.0.0.1:58235  
7189  
      1 0x00000001 RUN      0          1 127.0.0.1:58236  
7389  
      2 0x00000002 RUN      0          1 127.0.0.1:58238  
7489  
      3 0x00000003 RUN      0          1 127.0.0.1:58237  
7589  
      4 0x00000004 RUN      0          1 127.0.0.1:58239  
7289  
-----  
-----
```

Total(7189,7289,7389,7489,7589) Connected Clients = 5

\$\$\$ tmaxh9 (tmadm): ci -P

CLH 0:

```
-----  
-----  
cli_id   clid   status count  lastin_time  ipaddr  
usrname          tmax_port  
-----  
-----  
      7 0x00000007 QED      0          27 192.168.1.34:63307  
12399  
      8 0x00000008 QED      0          27 192.168.1.34:63308  
-----  
-----
```

```

12399
   9  0x00000009 QED      0      27  192.168.1.34:63309
12399
  10  0x0000000a QED      0      27  192.168.1.34:63310
12399
  11  0x0000000b QED      0      27  192.168.1.34:63311
12399

-----
-----
Total Connected Clients = 5
-----
-----

```

◦ 관련 이슈 : [IMS 108817](#)

• **discon 기능 추가**

- 16진수 clid로 discon 가능하다.
- 사용 방법

```

$$1 tmax1 (tmadm): ds [-h clhno] [-f] { -d clid }

```

항목	설명
{ -d clid }	해당 clid를 가진 클라이언트와의 연결을 해제한다.

- slog에 출력되는 값들 8자리 16진수로 수정하였다.
- MAXQCOUNT에 의한 예러에 clid 값 추가하였다.
- ci 결과로 qpcount, emcount가 출력되게 수정하였다.

항목	설명
qpcount	queue에서 purge된 메시지들의 개수이다.
emcount	queue 안에 있는 메시지들이 max 값에 도달한 횟수이다.

◦ 관련 이슈 : [IMS 147989](#)

• **rbs -o 옵션 추가**

- 서버 프로세스가 여러 개 기동되어 있더라도 old\_file에 해당하는 서버 프로세스들을 모두 suspend->down->cp->boot->resume->reboot 시키는 기능을 추가하였다.
- 사용 방법

```

$$1 tmax1 (tmadm): rbs [-o] new_file old_file

```

항목	설명
[-o]	old_file에 해당하는 서버프로세스들을 전부 종료시킨 후 new_file에 해당하는 서버 프로세스들로 모두 재기동시킨다.
new_file	새 파일명을 지정한다.
old_file	사용 중인 파일명을 지정한다.

◦ 관련 이슈 : [IMS 176487](#)

• **service 수행 시간을 microsecond 단위로 볼 수 있도록 기능 추가**

- tmaxin 상에서 microsecond 단위의 svc 수행 시간들을 출력하도록 기능추가 되었다.
- svclog, svcrcpt 에서 출력되는 svc 수행시간들을 소숫점 이하 6자리까지 표현되도록 기능추가 되었다.

◦ 관련 이슈 : [IMS 175917](#)

• **트랜잭션 처리 기능 추가 : txc / txr / txq [-t upper xid / lower xid]**

- 게이트웨이로부터 시작된 트랜잭션에 대해서 로컬에 pending이 존재하는 상황에서 장애에 의해서 로컬이 재기동한 경우 리모트에서 decision을 받아오지 못해 여전히 pending으로 남아 있고, 리모트에서도 재시도를 하지 않는 트랜잭션은 정리하기가 어려워 리모트로부터 시작된 pending 트랜잭션에 대해서 조회/commit/rollback 기능이 추가되었다.
- ID의 첫 부분은 upper xid, 두 번째 부분은 lower xid, 세 번째 xid는 bqual no이다.

▪ 조회 (txq -t)

```

$$1 tmaxi9 (tmaxin): txq -t
-----
XID                svgnno RXID                gwname
-----
00000000:00032088:00000000  3      05000403:00000001:00000000  gw1
00000000:00032089:00000000  3      00000102:00000001:00000000  gw1

```

▪ commit

```

$$4 tmaxi9 (tmaxin): txq -t
-----
XID                svgnno RXID                gwname
-----
00000000:0002d088:00000000  3      09090906:00000001:00000000  gw1
$$5 tmaxi9 (tmaxin): txc -t 0 2d088
XID 00000000:0002d088:00000000 svgnno 3.
Are you really want to commit?(n|y) y
txcommit completed. Please monitor by txquery -t 0 2d088

```

▪ rollback

```

$$4 tmaxi9 (tmaxin): txr -t 0 00032088
$$4 tmaxi9 (tmaxin): XID 00000000:00032088:00000000 svgnno 3.
Are you really want to roll back?(n|y) n

```

```
txrollback was canceled
```

- 관련 이슈 : [IMS 97235](#)

#### • maxrstart reset 기능 추가

- tmaxadmin에서 maxrstart 값을 reset이 가능하도록 tmaxadmin의 restat -r 옵션을 추가하였다.
- 예제

```
$ qpsx2@tmax1:/home/tmax1/tmax/sample/server>tmaxadmin -m
--- Welcome to Tmax Admin (Type "quit" to leave) ---

$$1 qpsx2 (tmaxadm): restat -v svr2 -r
Restarted count for SVR(svr2:4) cleared: 1
$$2 qpsx2 (tmaxadm):
```

- 관련 이슈 : [IMS 114095](#)

## 2.1.5. TDL

#### • tdlgetseqno()를 호출하는 경우 TDL의 Global seqno(마지막 tdlupdate 시간)가 아닌 가장 최신 시간을 채번해주는 기능 추가

- 각 노드(리전)별로 별도의 TDL 시스템을 사용하는 경우 tdlstart()가 호출된 상태에서 다른 region으로 tpforward, tpcall 등이 수행되면 callee region의 프로세스에서는 자신의 TDL 환경의 Global seqno가 더 최신이더라도 caller region의 TDL 환경의 Global seqno가 적용되면서 서로 정합성을 맞추기 어려워 질 수 있다.

tdl.cfg에 다음 옵션을 추가한다. (기본값: N)

```
GSEQNOTIME=[Y|N]
```

항목	버전
Y	채번할 때 호출 시점의 시간을 가져온다. 무조건 Global Seqno를 채번하는 것이 아닌 tdlgetseqno() 호출 시점의 시간과 서로 비교하여 더 큰 번호를 채번해주도록 하는 기능이다.
N	채번할 때 TDL의 Global Seqno를 가져온다. (기본값)

- 관련 이슈 : [IMS 140634](#)

#### • tdlclean 성능 향상 옵션 -f 추가

- 기존의 tdlclean의 동작 방식이 \$TDLDIR/run에 있는 라이브러리들을 일일이 nm을 통해 심볼들을 추출하여 TDL 공유 메모리에 있는 최신 라이브러리/함수와 비교를 하는 방식이어서 실행속도가 느린 문제점이 있었다.
- tdlclean -f 옵션을 사용하는 경우 run에 있는 라이브러리 정보를 힙 메모리에 모두 등록한 후 한번에 TDL

공유 메모리에 있는 최신 라이브러리의 함수들과 비교한다. -N, -m, -M 옵션과 함께 사용할 수 없다.

- 사용 방법

```
$ tdlclean -f
```

- 관련 이슈 : [IMS 138434](#)

- **TDL 환경설정 파일 및 mod,run 디렉터리를 멀티 노드에서 공유하는 기능 추가**

- 멀티 노드 환경에서 TDL을 공유 디스크 등을 이용해서 동일한 경로에서 공유하여 사용하는 경우를 위해 MODE=SHARED 기능을 추가하였다.
- tdl.cfg 환경설정에서 제공하는 옵션

```
MODE=SHARED
```

추가된 세부 옵션이다.

```
MASTERNODENO=0
```

MASTER로 동작할 노드 번호(0부터 시작), MODE=SHARED 옵션을 사용할 경우 반드시 설정해야 한다. 또한 TMAXDIR 환경변수를 지정하지 않으면 반드시 RACFILE 옵션을 설정해야 한다(Tmax 환경에서는 설정하지 않는다).

- SHARED 사용하는 경우 TDL 유틸리티의 기본 동작 방침
  - 모든 노드는 동일한 TDLDIR 환경변수 및 동일 위치의 tdl.cfg 환경설정 파일을 사용한다.
  - MASTER로 지정된 노드에서만 mod, run 디렉터리의 파일을 복사하거나 삭제한다.
  - 모든 노드에서는 자기만의 TDL 공유 메모리를 생성, 관리한다. 그러나 모듈의 버전 정보는 모든 노드가 동일하게 적용된다. (MODE=MASTER,SLAVE 동작과 동일)
  - TDL 공유 메모리에 대한 백업 파일을 강제로 생성, 관리한다.
- 관련 이슈 : [IMS 144058](#)

## 2.1.6. 도메인 게이트웨이

- **접속 시도 로그를 slog에 남지 않게 하는 CLOPT -x 옵션 추가**

- 환경 구성상 원격 연결이 끊어져 있는 도메인 게이트웨이에 주기적으로 접속 시도를 할 경우 slog에 대량의 로그가 남을 수 밖에 없게 된다. CLOPT 항목에 -x 옵션을 주었을 경우 해당 로그들을 ulog에만 남기고 slog에는 남기지 않도록 한다.
- 관련 이슈 : [IMS 117571](#)

## 2.1.7. Tuxedo 게이트웨이

- Tuxedo 7.2 이상의 프로토콜 지원

- Tuxedo 7.2 이상의 버전에서는 Tmax와 연동을 위해서는 [-t] 옵션을 적용해야 했다. TuxedoGW에 CLOPT -n 옵션을 지정하면 Tuxedo에서 -t 옵션을 적용하지 않아도 동작하는 기능을 추가하였다. 이 옵션을 지정하지 않으면 Tuxedo에서 -t 옵션을 반드시 지정해야 한다.
- 관련 이슈 : [IMS 97823](#)

## 2.1.8. JAVA 게이트웨이

- 접속 시도 로그를 slog에 남지 않게 하는 CLOPT -x 옵션 추가

- 환경 구성상 원격 연결이 끊어져 있는 JAVA 게이트웨이에 주기적으로 접속 시도를 할 경우 slog에 대량의 로그가 남을 수 밖에 없게 된다. CLOPT 항목에 -x 옵션을 주었을 경우 해당 로그들을 ulog에만 남기고 slog에는 남기지 않도록 한다.
- 관련 이슈 : [IMS 117571](#)

## 2.1.9. TCP 게이트웨이

- TCPGW 큐잉 기능 추가

- CLOPT 항목에 [-XFWDQ] 옵션을 주었을 경우 TCPGW로 tpforward될 때 TCPGW의 모든 CPC가 사용 중일 때 서비스가 큐잉되도록 기능이 개선되었다.
- 관련 이슈 : [IMS 101309](#)

- svclog 기능 추가

- CLOPT -i 옵션(-- 이전)을 주었을 경우 Tmax에서 리모트로 요청하고 리모트로부터 응답을 받는 요청에 대해서 서비스 로그를 기록하도록 기능 추가하였다.

TCS, UCS와 마찬가지로 5분 단위로 내용을 flush하며, TCPGW가 정상 종료될 때에도 미기록된 내용들을 flush시킨다.

TCPGW의 요청이 타임아웃 발생하는 경우에도 요청할 때부터 해당 발생시점까지 경과시간을 svclog에 기록한다. 이때 타임아웃 시간이 만약 10초로 지정되어 있었다면 TCS처럼 정확하게 시간을 체크하지 못하기 때문에 약간의 타임아웃 발생까지 오차가 있을 수 있음을 유의한다.

- 관련 이슈 : [IMS 101309](#)

## 2.1.10. Entera 게이트웨이

- entgw에서 broker negotiation 과정에서 hostname이 강제로 셋팅되는 것을 설정으로 변환

- entgw의 CLOPT -- 뒷부분 옵션 목록에 -h 옵션을 추가하였다.



옵션	설명
[-h user_hostname]	지정한 이름으로 hostname을 대체해서 broker 응답 메시지를 전송한다. 옵션 미지정시 기본값은 기존과 마찬가지로 gethostname() 호출 결과가 전송된다.

- 관련 이슈 : [IMS 101036](#)

## 2.1.11. RPC

### • tmaxrpc idlc에서 delphi xe 지원

- idlc -clang 옵션에 delphi xe 버전을 지원하는 delphixe 타입을 추가하였다.
- 사용 방법

```
$idlc -clang delphixe tstest.def
```

- 관련 이슈 : [IMS 97595](#)

## 2.1.12. Tcache

### • 로컬만 invalidate하는 옵션(-I)을 추가

- pfmtcacheadmin -I 옵션을 사용하는 경우 특정 로컬 TCache의 데이터를 초기화하는 기능을 추가하였다.
- 사용 방법

```
$ pfmtcacheadmin -I [cacheName]
```

- agent svc invalidate 시 local invalidate의 성공, 실패 여부 메시지를 남기도록 수정하였다.

- 관련 이슈 : [IMS 121387](#)

### • libpfmTCacheWrap.so 라이브러리를 추가

- 기존 libpfmTCache.so 외에 wrapper function을 포함하고 있는 libpfmTCacheWrap.so 라이브러리를 추가하고 여기에 고객사의 wrapper function을 추가하였다.
- 추가된 wrapper function의 목록은 다음과 같고 이들은 각각 아래의 함수를 wrapping 한다.

```
long pfmTCacheGetItem
long pfmTCacheGetAllItems
long pfmTCacheReplaceItem
long pfmTCachePutItem
long pfmTCacheAppendItem
long pfmTCacheInvalidateItem
long pfmTCacheInvalidateLocalItem
long pfmTCacheInvalidateAllLocalItems

pfmTCacheGet
```

```

pfmTCacheGetAll
pfmTCacheReplace
pfmTCachePut
pfmTCacheAppend
pfmTCacheInvalidate
pfmTCacheInvalidateLocal
pfmTCacheInvalidateAllLocal

```

- wrapper library를 적용하는 방법은 TCache를 설치한 뒤 libpfmTCache.so 라이브러리를 백업하고 libpfmTCacheWrap.so 파일의 이름을 libpfmTCache.so로 변경한다.
- 관련 이슈 : [IMS 116106](#)

## 2.1.13. 기타

- 클라이언트가 종료되었을 때 CustomGateway 서버에서 이를 감지하고 후속 작업을 할 수 있는 API 제공한다.
  - 프로토타입

```

typedef enum {
CWRC_END,      /* client가 tpend()를 호출하고 종료함 */
CWRC_ABEND,    /* client가 비정상 종료함 */
CWRC_INVALID, /* 입력한 clid가 잘못되어 있어서 감지 안함 */
CWRC_NOWATCH, /* clh의 메모리 부족으로 인해 clid 감지 안함 */
CWRC_NORMAL   /* 사용하지 않음 */
} cwrc_e;
typedef void (*CliWatcherCallback)(int clid, cwrc_e reason, void *args);
typedef int (*SendToCLHCallback)(int clhno, header_t *hp, void *args);
void tmaxgw_set_sendtoclh_cb(SendToCLHCallback callback);
int tmaxgw_unmatched_msg_process(header_t *hp);
int tmaxgw_setcliwatcher(int clid, CliWatcherCallback callback, void *args, int flags);
int tmaxgw_clrcliwatcher(int clid, void **pargs, int flags);

```

- 관련 이슈 : [IMS 96544](#)
- **TMAX\_DEBUG 지정 시 로그 파일 및 터미널 화면에 출력하는 기능 추가**
  - TMAX\_DEBUG 지정된 상태에서 환경변수 TMAX\_ERR\_STDOUT=Y 를 설정하면 로그파일에도 기록하고, 터미널 화면에도 출력한다.
  - TMAX\_DEBUG 환경변수에 지정한 로그 파일 이름 뒤에 날짜를 표기하여 파일을 생성하며, 로그 파일 이름 끝에 ".pid"가 있으면 해당 부분을 수행하는 프로세스의 pid 로 변경하여 파일 이름을 생성한다.
  - 대상 : 클라이언트, TDL 관련 실행파일, Tmax utility programs
  - 관련 이슈 : [IMS 114552](#)

## 2.2. 변경 기능

본 절에서는 변경된 주요 기능에 대해 설명한다.

## 2.2.1. ENGINE

- **tmdown -g나 tmdown -A를 시작할 때부터는 새로운 서버가 접속하지 못하도록 수정**
  - tmdown -g나 tmdown -A로 특정 서버 그룹이나 서버들에 대해서 종료하는 순간 asqcount나 서버 비정상 종료로 인해서 새로운 서버가 기동할 경우 정상적으로 tmdown이 되지 못하는 현상을 수정하였다.
  - 관련 이슈 : [IMS 98268](#)
- **동일 NODE 내에서 2개 이상의 SVRGROUP을 COUSIN으로 설정할 때 tmdown / tmboot를 수행할 때 첫 번째 svri의 서버(primary 서버 그룹)만 기동/종료되는 현상 수정**
  - tmboot -S 옵션은 동일 노드 내 COUSIN 서버 그룹의 동일 서버 모두 각각 MIN 개수만큼 기동한다. tmdown -S 옵션은 동일 노드 내 COUSIN 서버 그룹의 동일 서버 모두가 종료된다.
  - 서버 라이브러리의 tmdadmin() API의 TMADM\_BOOT, TMADM\_DOWN 명령과 관련된 스펙을 변경하였다.
    - tmboot/tmdown의 -S 옵션과 같은 의미의 TMADM\_USFLAG가 추가되었다.
    - TMADM\_SFLAG 대신 TMADM\_USFLAG를 사용할 경우에는 struct tmdm\_boot의 args.count에 값을 지정하지 않는다.
    - TMADM\_USFLAG와 TMADM\_GFLAG를 함께 지정하는 경우 해당 서버 그룹의 서버들만 적용된다.
  - 다음은 tmdadmin()의 TMADM\_BOOT, TMADM\_DOWN에서 사용가능한 flags 조합이다.

flags	설명
TMADM_SFLAG	-s 옵션과 동일하다. count를 지정해야 한다.
TMADM_SFLAG   TMADM_GFLAG	-s -g 옵션과 동일하다. count를 지정해야 한다.
TMADM_TFLAG	-t 옵션과 동일하다. count를 지정해야 한다.
TMADM_USFLAG	-S 옵션과 동일하다.
TMADM_USFLAG   TMADM_GFLAG	-S -g 옵션과 동일하다.
위의 모든 경우   TMADM_FFLAG	-i 옵션과 동일하다. TMADM_DOWN에서만 사용이 가능하다.

- 관련 이슈 : [IMS 79295](#)
- **MAXQCOUNT가 1 미만의 값일 경우 오류가 발생하도록 수정**
  - MAXQCOUNT를 tmconfig 환경 설정이나 tmdadmin의 동적 설정을 통해 설정할 때 1 미만의 값을 입력하면 오류가 나도록 변경하였다.
  - 관련 이슈 : [IMS 125009](#)
- **CLH2053, CLH2052 에러 메시지 개선**
  - CLH에서 연결이 끊긴 서버에서 들어온 메시지를 버릴 때 나타나는 로그 CLH2053에 msgtype과 spri를 같이 출력하도록 변경하였다.

- 이전

```
2053 msg discarded due to closed server connection : svc = %s
```

- 이후

```
2053 msg discarded due to closed server connection : svc = %s, msgtype = %d, spri = %d
```

- CLH2052도 msgtype을 출력하도록 변경하였다.

- 이전

```
2052 msg discarded due to closed client(%#x) connection : svc = %s
```

- 이후

```
2052 msg discarded due to closed client(%#x) connection : svc = %s, msgtype = %d
```

- 관련 이슈 : [IMS 116465](#)

- **OPENINFO 절 암호화 사용 시 아무것도 입력하지 않아도 에러가 발생하지 않도록 수정**

- 관련 이슈 : [IMS 124497](#)

- **ASQCOUNT가 설정되었을 때 MAXQCOUNT를 0으로 설정 못하도록 변경**

- 기존에는 매뉴얼대로 MAXQCOUNT는 1 이상으로만 설정 가능했으나, 큐잉을 하나도 하지 않고 바로 에러를 리턴하고 싶은 경우를 위해 설정을 변경하였다.

- 관련 이슈 : [IMS 179576](#)

- **TMMOPT -S, D 옵션 동작 방식 변경**

- 기존에 -D 옵션은 기본 1초 동안 자신의 main과 연결되지 않으면 backup svr를 기동하게 하고 있었다. 기존에 -S 옵션은 -D 옵션과 같이 사용해야만 동작했다.

-D와 -S 옵션 분리해 -D 옵션만 설정했을 때는 자동으로 backup svr가 기동하지 않고, -S 옵션을 지정하면 해당 초단위 시간이 흐른 뒤에도 main svr가 떠 있지 않으면 backup svr 기동하도록 수정하였다.

- select에 주어지는 시간은 NLIVEINQ와 -S 옵션 값중 더 작은 수가 배정되므로, NLIVEINQ값이 S 옵션 인자값보다 작을 경우 S값으로 지정해 준 초만큼 지난 후에 backup svr가 기동하는 것이 아닐 수 있다.

- 관련 이슈 : [IMS 179576](#)

- **tmboot를 수행할 때 TLM, TMS의 복구 시간이 오래 걸리는 현상 개선**

- TLM에서 TXLOG, GWTXLOG 파일을 복구하는데 많은 시간이 걸리는 부분을 개선하였다.

- TMS에서 RM에게 xa\_recover()를 호출할 때 한 번에 10개씩 요청하는 부분을 기본 256개씩 요청하도록 변경하였다. 이 값은 CLLOPT 항목에서 -c <count> 옵션으로 변경 가능하다. 기본값은 256이며, 최소 10, 최대 8192까지 설정 가능하다.

◦ 관련 이슈 : [IMS 105295](#)

• **멀티 백업 우선순위 개선**

- 백업 서버 기동순서를 BACKUP에 정의한대로 수행하되 앞선 노드가 살아있으면 백업 서버를 기동하지 않도록 수정하였다.
- TMMOPT "-S sec"로 설정하는 경우 tmm을 기동할 때 자신의 노드가 백업 설정상의 원본 서버 그룹이 아니더라도 모든 노드와 연결시도를 해보고, -S 설정한 시간이 지나도 자신보다 백업 순서가 우선하는 노드가 NRDY일 경우 자신의 노드에서 백업 서버를 기동한다.
- TMMOPT -D 옵션(auto fallback)에 대해서 앞선 노드가 살아나면 해당 노드에 대해서 자신이 수행하고 있는 백업 서버를 모두 종료하도록 수정하였다.
- 서버 프로세스 기동 중에 SVR\_DOWN\_REQUESTED, MAX\_SPR\_REACHED 등의 에러가 발생하고 나서 boot timeover 에러가 발생하는 현상 수정하였다.
- DEBUG1 레벨로 지정된 epoll log를 DEBUG4 레벨로 변경하였다.
- 관련 이슈 : [IMS 172747](#)

• **epoll log 로그레벨 변경**

- DEBUG1 레벨로 지정된 epoll log를 DEBUG4 레벨로 변경하였다.
- 관련 이슈 : [IMS 172747](#)

• **tmadmin 등이 tmm에 접속 에러 시 구체적인 원인 설명**

- tmadmin, tmboot, tmdown, libsvr, libtms 등 모든 tmm 접속 프로그램에서 접속 에러가 발생하는 경우 에러 원인 설명이 없었다.
- 접속 실패시 다음과 같이 구체적인 원인이 사용자 콘솔에 출력될 수 있도록 수정하였다.

```
(E) ADM2045 failed to register to TMM : ADM_MAIN_STARTED [COM0615]
(F) ADM3010 failed to register to TMM [ADM0015]
```

◦ 관련 이슈 : [IMS 100280](#)

## 2.2.2. 클라이언트/서버

• **libsvr \*, libtms \*에서 openssl 라이브러리 함수/변수명 변경**

- 다음과 같이 라이브러리 함수/변수명이 변경되었다.

변경 전	변경 후
RC4	_lspf_RC4
RC4_options	_lspf_RC4_options
RC4_set_key	_lspf_RC4_set_key
RC4_version	_lspf_RC4_version

◦ 관련 이슈 : [IMS 115833](#)

- **tpstart를 하지 않은 상태에서 특정 함수들을 호출할 경우 출력 상태 변경**

- tpstart()를 호출하는 경우 그 이전에 호출하여 설정했던 트랜잭션 동작에 대한 정보가 계속 유지되도록 변경되었다.

- 0이 리턴되는 함수

```
tx_info
tx_set_transaction_timeout
```

- 정상처리되는 함수

```
tx_info
tx_set_transaction_timeout
```

- 관련 이슈 : [IMS 114518](#)

- **fbinit(), fbput()를 호출할 때 FBEVTSYNTAX(bad viewfile) 발생 현상으로 인한 동작 변경**

- tpalloc("CARRAY")로 할당한 버퍼를 fbinit() 호출하면 FBEVTSYNTAX(16) 에러를 리턴하는 문제가 발생하였다.

Tmax 5부터는 FDL 함수를 사용할 때 tpalloc("FIELD")로 할당한 버퍼만 사용할 수 있도록 개선되었는데 Tmax 4에서는 이러한 버퍼 타입 체크 로직이 수행되지 않으며 버전 업그레이할 때 문제가 되고 있다.

- CARRAY 타입 등 FIELD 타입 이외의 버퍼를 fbinit() 호출하는 경우 해당 버퍼의 유형을 FIELD 타입으로 변경하였다.
- FIELD 타입 이외의 버퍼를 FDL 함수에 사용할 때 Tmax 4.0 호환을 위해 에러 처리 로직을 제거한다. 즉, 에러를 발생시키지 않고 FDL 함수가 수행되도록 한다. 대신 잠재적 위험이 있기 때문에 다음 에러 로그를 프로세스 당 최초 1회만 출력한다.

```
CLI0248" "SVR0248 buffer type error(CARRAY). use FIELD type. [FDL0515]
```

- TMAX\_TRACE를 설정할 때 남겨지는 로그에서 FBEVTSYNTAX 대신 FBESETVER로 수정하였다.

- 관련 이슈 : [IMS 112081](#)

- **tpspracall 함수의 큐잉 및 TPBLOCK flag 기능 지원**

- tpspracall이 큐잉되도록 하고, flag의 TPBLOCK 기능이 동작하도록 개선하였다.
- 클라이언트가 연결한 CLH가 맞는데 spr이 NOT\_READY인 경우 기존에는 TPENOENT를 리턴해 주고 있었으나, 이를 TPEOREADY로 변경하였다.
- MULTICLH=N인 서버는 tpspracall() api는 지원하지 않으며, 클라이언트가 연결한 clhi와 요청한 spr의 clhi를 비교하여 맞지 않을 경우 기존에는 TPENOENT를 리턴해주고 있었으나 TPEPROTO 에러를 리턴하도록 변경하였다.

- 관련 이슈 : [IMS 145674](#)

### 2.2.3. 관리 도구

- **gwi, txgwi, ntxgwi, jgwi 등의 명령을 수행하면 Hang이 걸리는 현상 개선**
  - 게이트웨이(Domain GW, JEUS GW, TUXEDO GW)가 기동 도중에 Listen해야 하는 포트를 다른 프로세스가 선점하고 있어서 Listen이 실패하는 경우에 이 상태에서 tadmin에서 gwi, txgwi, ntxgwi, jgwi 등의 명령을 수행하면 Hang이 걸리는 현상이 있다.
  - Listen port를 사용하기 위해 대기하는 동안 TMM으로부터 들어오는 요청을 받아서 처리할 수 있도록 개선하였다.
  - 관련 이슈 : [IMS 111191](#)
- **ci -P 명령어를 통해 qpcount와 emaxcount 정보를 받아오는 옵션 수정**
  - ci -P 명령어를 통해 qpcount와 emaxcount 정보를 받아오도록 옵션을 수정하였다.
  - 사용 방법

```
$$1 tmax1 (tadm): ci [-P]
```

항목	설명
[-P]	포트 번호 정보와 qpcount, emaxcount 정보를 추가로 출력한다.

- 관련 이슈 : [IMS 153093](#)

### 2.2.4. 유틸리티

- **tadmin에서 libtermcap 대신 libncurses를 참조하도록 수정**
  - tadmin에서 libtermcap 대신 libncurses를 참조하도록 수정하였다. (Linux용 ncpu 64bit 빌드)
  - 관련 이슈 : [IMS 169868](#)
- **tmboot -s svr -k ksvr에서 ksvr를 MAX까지 기동시킬 수 있도록 개선**
  - tmboot -s -k를 수행하는 경우 TMM에 의해 추가 기동된 spr가 존재한다면 MAX를 넘지 않는 선까지 추가 기동하도록 개선하였다.
  - tmboot -S를 수행하는 경우 min 개수만큼 spr 기동시키도록 개선하였다. (불가할 경우 max까지는 보장)
  - 관련 이슈 : [IMS 164518](#)
- **컴파일러 없이 tmaxlibver 사용하도록 개선**
  - nm으로 ProdVer로 시작하는 심볼이 존재하는지 체크하여 해당 정보를 출력하도록 개선하였다.
  - 사용 방법

```
$ tmaxlibver -l libsPfmAgent.so -6  
libsPfmAgent.so for TMAX Version 6.0 Sp #0 Fix #1 r11552 fd 32768 64bit Aix71 patch
```

- 관련 이슈 : [IMS 119149](#)

## 2.2.5. TDL

- **TDL MODE=SHARE로 설정 시 멀티 노드 환경에서 일부 노드가 죽어도 tdlupdate가 성공하도록 개선**
  - MASTERNODENO로 지정한 노드가 내려가 있으면(racd가 기동되지 않을 경우도 포함) Tmax 환경설정의 NODE 절 순서상 다음 번호의 노드가 마스터 노드의 역할을 대신 하도록 개선하였다.
  - tdl.cfg 파일에 LOGDIR 옵션을 추가해서 TDL 유틸리티의 로그들을 지정한 파일에 기록한다.
  - 관련 이슈 : [IMS 166178](#)

- **tdlupdate 성능 개선**

- tdlupdate를 수행할 때 대상 파일에 대한 access가 여러 포인트에서 추가되어 기존 대비 2배 느려지는 현상이 발생하여 access를 최소화하도록 수정하였다.
- backup file update 방식을 기존에는 라이브러리 내의 함수 하나마다 매번 backup file update를 개별적으로 실행했다. 이를 한번에 처리하도록 개선하였다.
- TDL 공유 메모리에 정보를 반영하는 시점에는 SIGKILL, SIGSTOP, SIGCONT를 제외한 모든 시그널에 대해서 중단되지 않도록 signal block 기능을 추가하였다.

위 항목 외의 시점에 <Ctrl>+C 등의 시그널을 받으면 파일 복사 등의 과정을 모두 원래대로 복원시킨다.

- 대상 시그널 : SIGHUP, SIGINT, SIGQUIT, SIGTERM, SIGUSR1, SIGUSR2
- 관련 모듈 : tdlclean / tdlupdate
- copy 방식을 기존의 /bin/cp를 이용하는 방법 외에 자체적으로 복사하는 기능을 추가하였다.

tdl.cfg에 다음 옵션을 추가한다. (기본값: mmap)

```
COPY=[cp|mmap]
```

항목	버전
cp	/bin/cp를 이용하는 방법이다.
mmap	tdlcall 내부적으로 파일 복사를 메모리 맵 방식으로 복사하는 것으로 cp에 비해서 약간의 성능상의 이점이 있다.

- 관련 이슈 : [IMS 104752](#)

## 2.2.6. Tcache

- **TDL 의존성 제거 및 wait 모드로 Lock 대기 시 usleep 쓰는 부분을 timedlock으로 변경**

- TCache를 Tmax 버전마다 다른 버전으로 제공하면 관리하기 쉽지 않으므로, 하나의 버전으로 Tmax에 대응할 수 있도록 수정하였다.
- TCache Get, Put TCACHE\_FLAG\_WAIT flag로 호출시 pthread\_rwlock\_trylock 호출 후 실패하는 경우 usleep(1000)으로 최대 1000번 반복 시도하는 부분을 pthread\_rwlock\_timedlock()으로 변경하고 최대 대기시간은 기본값 1초(기본과 동일)이며, LOCK\_WAITTIME 환경설정을 통해 1~86,400,000 밀리초까지 설정하여 대기시간을 설정할 수 있다.



- 관련 이슈 : [IMS 116106](#)

## 2.2.7. 도메인 게이트웨이

- **CLOPT -p 옵션의 메시지 전문 출력을 LOGLVL별로 지정할 수 있도록 기능 변경**

- 기존에는 CLOPT -p 옵션 및 LOGLVL=DEBUG4을 설정하면 거래요약정보 및 전문을 모두 출력했다.

해당 버전 이후 DEBUG1로 설정하는 경우 전문을 제외한 한 줄만 로깅되고, DEBUG2 이상 설정하는 경우 전문을 포함하여 로깅할 수 있도록 기능을 변경하였다.

- 관련 이슈 : [IMS 114185](#)

## 2.2.8. Tuxedo 게이트웨이

- **접속된 TUXGW로 접속이 시도된 경우 해당 접속의 IP, Port를 포함한 에러 로그 남기도록 개선**

- tuxgw log

```
(E) GATEWAY3008 overlapped connection from 192.168.15.116:48052 [TUXGW0162]
(E) GATEWAY3010 connection error from remote gateway [TUXGW0002]
```

- 관련 이슈 : [IMS 108835](#)

- **tuxgw 접속 종료 시 에러 로그에 포트 번호 추가 및 ulog에 시간 출력 개선**

- tuxgw log

```
(E) GATEWAY2062 remote gateway closed: 192.168.5.142:PORT [TUXGW0307]
```

- ulog에 출력되는 slog 메시지에서 시간이 출력되도록 개선하였다.

- 관련 이슈 : [IMS 108835](#)

## 2.3. 버그 패치

### 2.3.1. ENGINE

- **UCS 서버 추가 기동할 때 outbound cpc로 스케줄링되는 현상 수정**

- 서버가 추가 기동되면서 cpc를 등록하는 과정에서 cpc 채널이 등록이 완료된 순간 현 시점에 큐잉된 것을 꺼내서 스케줄링한다. 추가로 등록하는 cpc 채널이 인바운드인 경우 문제가 없지만, 아웃바운드일 경우 처리가 안되는 현상이 있었다. 인바운드와 아웃바운드를 구분하여 스케줄링하도록 수정하였다.

- 관련 이슈 : [IMS 103483](#)

- **TMAXGW 비정상 종료될 때 클라이언트에게 잘못된 메시지 타입을 전달하는 오류 수정**

- TMAXGW가 비정상 종료되었을 때 클라이언트에 XA\_COMMIT, XA\_PREPARE의 에러 응답 메시지가 전송되어 다음의 에러가 발생하는 문제가 있다.

```
CLI 3039 internal error : unknown message type :2112 [CSC5713]
```

- XA 관련 메시지는 CLH와 TMS 혹은 트랜잭션 지원 게이트웨이 간에 오고 가는 메시지 타입으로 CLH가 트랜잭션 게이트웨이와의 연결이 끊겼을 때 XA 메시지 타입이면 클라이언트에게 응답을 주지 않도록 수정하였다.
- 관련 이슈 : [IMS 103569](#)

- **booting time over가 발생하는 경우 서버명이 unknown으로 출력되는 현상 수정**

- boottime 계산을 위해 내부적으로 index를 관리하는데, cfgadd로 서버가 추가되는 경우 대해서는 index 값이 초기화되지 않은 문제를 수정하였다.
- 관련 이슈 : [IMS 107128](#)

- **TMSTYPE=STD\_MT를 설정할 때 오류 수정**

- TMSTYPE=STD\_MT를 설정할 때 다음의 에러가 발생하는 오류를 수정하였다.

```
CLH2131 invalid thri: 4
```

- STD\_MT 타입으로 지정하면 tx\_commit(), tx\_rollback() 요청이 TMSTHEADS 이상 발생할 때 CLH2131 에러가 발생하고 클라이언트 또는 서버로 응답이 전송되지 않는 문제를 수정하였다.
- 동시에 TMS로 요청하는 건수가 많아지는 경우에 갑자기 TMS로 더 이상 요청이 전달되지 않는 문제를 수정하였다.
- slog에서 thread id가 음수로 나오는 문제를 수정하였다.
- 여러 번 XA 요청이 수행된 이후에 Tmax를 정상적으로 tmdown할 때 TMS가 비정상 종료되는 문제를 수정하였다.
- 관련 이슈 : [IMS 107299](#)

- **기동 중인 서버가 없을 때 CLL이 비정상 종료로 재기동되면 발생하는 문제 수정**

- 기동 중인 서버가 없을 때 CLL이 비정상 종료로 재기동되면 CLH와 연결을 맺지 못하는 현상을 수정하였다.
- 관련 이슈 : [IMS 108258](#)

- **클라이언트에서 MultipleRM 서비스를 트랜잭션으로 묶어 호출하는 경우 xa\_start 에러 발생 수정**

- 클라이언트에서 MultipleRM 서비스를 트랜잭션으로 묶어 32번 초과 호출(tpcall)하면 서비스를 호출할 때마다 bqno를 증가시킨다. Oracle의 경우 bqno가 32개 초과일 경우 xa\_start 에러가 발생한다.

서비스를 호출할 때 bqno를 무조건 증가시키지 않고, 필수적인 경우에만 증가하도록 수정하였다. (tpcall만 해당)

- 관련 이슈 : [IMS 108631](#)

- **SVRQTIMEOUT 기능을 사용할 때 잘못된 에러가 출력되는 현상 수정**

- CLH0051 에러만 발생해야 하는데 CLH0922 에러가 발생하는 현상을 수정하였다.

- 동적설정으로 timeout 0으로 off시킨 이후에 svrplib에서 무조건 EQPURGE 에러 발생하는 현상을 수정하였다.

- 관련 이슈 : [IMS 111648](#)

- **멀티 노드 환경에서 TMM 비정상 종료 현상 수정**

- 네트워크 지연 등으로 상대 TMM과 연결을 확인하는 과정에서 잘못된 메모리를 참조하여 TMM이 비정상 종료되는 문제를 수정하였다.

- 관련 이슈 : [IMS 112078](#)

- **tmdown -g svgname 후 POD 서버를 호출할 때 기동하지 않는 문제 수정 및 -b 기능 추가**

- POD 형태의 서버가 tmdown -g svgname 시점에 프로세스가 기동되어 있지 않다면, 서비스 요청이 들어와도 기동하지 않는 문제를 수정하였다.

- POD 형태의 서버가 tmdown -b -g svgname 시점에 프로세스가 기동되어 있지 않다면, 서비스 요청이 들어오면 기동하지 않도록 기능을 추가하였다.

- 관련 이슈 : [IMS 112124](#)

- **ENVFILE에 환경변수 설정 후 tmboot로 서버를 기동할 때 tmboot가 비정상 종료되는 현상 수정**

- 기동 과정에서 ENVFILE의 변수를 fgets로 한 번에 읽어 들이는 버퍼의 길이가 1024이고, 읽은 후 \${ 환경변수 패턴이 존재하면 해당 값을 치환하는 과정에서 치환된 최종 문자열 길이가 버퍼 크기를 초과하면서 다른 변수(fgets에 쓰이는 file pointer)를 깨트리는 문제가 있다.

- 환경변수 path 길이를 기존 1024byte에서 65536byte(=64KB)로 변경하였다. 단, 플랫폼이나 컴파일 최적화에 따라 64KB보다 클 수 있다.

다음의 경우에 적용된다.

- ENVFILE 항목에 file을 설정한 후 tmboot로 일반 서버가 기동
- ENVFILE 항목에 file을 설정한 후 tmaxadmin rst로 일반 서버가 기동
- ENVFILE 항목에 file을 설정한 후 tmm로(재기동) 일반 서버가 기동
- ENVFILE 항목에 file을 설정한 후 tmboot로 TCPGW 서버가 기동
- tmaxreadenv(file)
- racd -i file

- 관련 이슈 : [IMS 117013](#)

- **TMM으로 비정상적인 메시지가 들어올 경우 TMM이 비정상 종료하는 현상**

- tmax에서 생성된 메시지가 아니어서 헤더 부분(길이)을 참조하여 메모리 핸들링할 때 비정상적으로 메모리를 참조하여 Core가 발생하는 문제를 수정하였다.

- 관련 이슈 : [IMS 90293](#), [IMS 118729](#)

- **cfgadd 메시지가 다른 메시지와 순서가 섞일 경우의 오동작 문제 수정**

- cfgadd가 실행되고 있을 때 다른 여러 가지 메시지가 TMM과 CLH 사이에 전송이 되는 경우 타이밍에 따라 엔진에서 블록 현상이 일어나고 magic number error가 발생하는 문제가 있어 메시지가 명확히 구분 처리되도록 수정하였다.

- 관련 이슈 : [IMS 136977](#)

- **백업 설정 서버가 중복되서 기동되는 문제 수정**

- 이미 백업 서버가 기동 중임을 나타내는 상태를 별도로 관리하여 문제를 수정하였다.

- 관련 이슈 : [IMS 136637](#)

- **Active & Backup 상황에서 Auto Fail-Back 적용 또는 미적용시 거래 장애 발생 수정**

- active 노드의 fail-back 에러코드를 출력하도록 간헐적으로 backup 노드의 서버들이 자동으로 종료되지 않는 현상이었다. backup 노드에서 기동 중인 active 노드로 먼저 연결이 되는 비정상 케이스에서 active 노드가 종료될 수 있도록 수정하였다.

- 관련 이슈 : [IMS 148443](#)

- **tmm log thread 사용 시 비정상종료 현상 수정**

- main-thread와 통신 부분에서 인터럽트 받을 경우 재처리하도록 수정하였다.

- NULL 포인터 체크하여 로깅을 수행 못하도록 수정하였다.

- 관련 이슈 : [IMS 142910](#)

- **CLH에서 다른 노드의 CLH에게 서버 기동을 알릴 때 Core 발생 현상 수정**

- clh의 패치 버전을 확인하는 부분에서 이미 free한 msgtype으로 검증을 시도하는 부분을 수정하였다.

- 관련 이슈 : [IMS 151131](#)

- **멀티 백업 환경에서 main이 죽지 않았음에도 불구하고 backup이 수행되는 현상 수정**

- backup\_startup\_timeout 기본값을 0으로 수정 후 -D 옵션이 켜져야만 값을 1로 바꾸도록 수정하였다.

- 관련 이슈 : [IMS 151131](#)

- **TMM 서버 재기동 시에 restart count 초기화되지 않는 문제 수정**

- 다음의 경우 모두 초기화를 수행할 수 있도록 수정하였다.

- SERVER 절 CLOPT -r 옵션을 사용하면 서비스 타임아웃으로 서버가 TPEXIT로 종료되면 restart count 초기화

- gperiod가 지나면 restart count를 초기화

- tmdown -S 수행 시 초기화

- maxrstart까지 restart count가 증가할 경우 초기화

- 관련 이슈 : [IMS 101667](#)

- **CUSTOM GATEWAY 서버 재기동할 때 발생하는 문제 수정**

- CUSTOM GATEWAY 서버가 재기동할 때 다음의 에러가 발생하는 문제를 수정하였다.

```
TMM3033 svr (*) booting time (10) is over
```

- 관련 이슈 : [IMS 112051](#)

- **TLM이 TXLOG 파일을 복구할 때 마지막 XID 채번 오류 수정**

- Tmax가 재기동하면 TLM은 TXLOG 파일에 데이터가 존재하면 마지막 XID + 10240 번호를 CLH에게 전달하고, 데이터가 존재하지 않으면 현재 시간 % 0xFFFF 모듈러 연산값을 초기값으로 CLH에게 전달한다. 그리고 CLH는 이 시작값부터 트랜잭션을 요청할 때 XID를 순차적으로 발급한다.

위 과정 중 TLM이 마지막 XID 채번 과정에서 항상 현재시간 % 0xFFFF 값을 전달하는 버그가 발생해서

중복된 XID로 존재하게 되어, 복구 과정 중 실제 decision은 롤백이나 며칠 전 검색된 XID는 decision이 커밋이라서 실제 decision은 롤백이나 커밋이 수행되는 현상이 발생하는 오류를 수정하였다.

- 관련 이슈 : [IMS 116341](#)

#### • suspend -s 시 asqcount 증가 안하도록 수정

- suspend -s로 서비스를 막았을 경우 CLH에서 블록된 해당 서비스에 의해 asqcount가 증가하지 않도록 수정하였다.
- suspend -s로 서비스를 막았을 경우 해당 서버가 종료되어도 블록된 정보를 유지하도록 변경하였다.
- 관련 이슈 : [IMS 138006](#)

#### • 동적으로 mutliclh=N인 서버 추가 시에 clh0에만 접속하는 현상 수정

- 추가된 서버의 mutliclh, server의 CLH별 maxsvr 값이 초기화되게 수정하였다. 이 외에도 tadmin ci를 수행하는 경우 Core가 발생하면서 모든 접속된 클라이언트가 끊겨버리는 현상을 수정하였다.
- 관련 이슈 : [IMS 151341](#)

#### • CONV 서버로 tpsprcall() 호출 시 응답받지 못하는 현상 수정

- CONV=B 옵션은 대화형 채널 요청을 한 서버가 다른 서버로 넘겨줄 때를 가정하고 패치한 것으로 서버간에 대화형 채널을 넘겨줄 때는 tpforward를 반드시 호출 할 것을 가정하고 있다.

따라서, CONV = O (OPEN)을 추가해 서버 라이브러리에서 CONV=O인 경우 tpsprcall(), tpconnect() 요청에 따라 tpreturn()를 수행하는 경우 정상적으로 처리되도록 수정하였다.

또한 CONV=Y인 설정에서 tpsprcall()이 리턴되더라도 정상적으로 TPEPROTO 에러가 발생하도록 수정하였다.

- CONV=B 설정에서 tpconnect()가 안되는 현상을 수정하였다.
- 관련 이슈 : [IMS 152775](#)

#### • CLH 간 cousin server 기동 시 ready notify에 대한 slog 기록 방지 루틴 추가

- 메시지 타입을 CLH의 버전을 체크해서 수신 가능한 CLH에게만 전송하도록 수정하였다.
- 관련 이슈 : [IMS 85842](#)

#### • clh losssvc 호출 시 비정상 종료 현상을 수정

- clh losssvc를 호출하는 경우 비정상 종료 현상을 수정하였다.
- 관련 이슈 : [IMS 101112](#)

#### • TSM에서 서비스가 실제 서비스 타임이 지나지 않았는데도 kill을 하는 현상 수정

- TSM에서 서비스가 실제 서비 스타임이 지나지 않았는데도 kill을 하는 현상을 수정하였다.
- 관련 이슈 : [IMS 123310](#)

#### • Cousin Svrgroup의 tmdown 요청 이후에 Extern Server에서 프로토콜 변경으로 인한 오동작 수정

- TMM에서 구버전 프로토콜로 요청이 들어올 경우 이를 체크해서 최대한 호환성을 제공하도록 수정하였다.
- 관련 이슈 : [IMS 128417](#)

#### • 암호화, 압축 사용 시 송신 데이터 크기가 4KB 이상일 경우 클라이언트 연결이 종료현상 수정

- 암호화, 압축을 사용하는 경우 송신 데이터 크기가 4KB 이상일 경우 클라이언트 연결이 종료되는 현상을

수정하였다.

- 관련 이슈 : [IMS 103735](#)

- **멀티 노드에서 internal error 에러 메시지 발생하는 현상 수정**

- TMM에서 네트워크 통신 이벤트를 관리하는 버퍼를 생성하는 경우 write 소켓 관련 이벤트를 처리하는 부분이 초기화가 안되어 있던 문제를 수정하였다.
  - 관련 오류 메시지

```
(E) TMAX00124 internal error : invalid event flag(8ca0010d) [EVT0007]
```

- 관련 이슈 : [IMS 143719](#)

- **AIX에서 TMM이 무한루프를 도는 현상 수정**

- pollset\_ctl 함수 호출에 실패하는 경우 실패가 EAGAIN이나 EINTR에 의한 것이 최대 5번까지 재시도하며 아래 로그가 발생하도록 수정하였다.

```
(E) TMM0209 internal error : pollset_ctl failure
```

- 관련 이슈 : [IMS 154357](#)

- **Linux에서 TMMOPT -g 옵션 설정 시 TMM CPU가 100%되는 현상 수정**

- Linux에서 TMMOPT -g 옵션을 설정하는 경우 TMM CPU가 100%되는 현상을 수정하였다.
- 관련 이슈 : [IMS 172041](#)

- **트랜잭션이 시스템에 의해 rollback된 상태에서 트랜잭션이 처리 오류 수정**

- 트랜잭션이 시스템에 의해 rollback된 상태에서 트랜잭션에 포함된 다른 서비스가 요청될 때 다른 XID로 트랜잭션이 처리되는 현상을 수정하였다.
- 관련 이슈 : [IMS 175850](#)

- **CLH가 완전히 기동하지 않은 상태에서 클라이언트가 접속을 시도하는 경우 비정상 종료하는 현상 수정**

- CLLCONNLB = "LC" 설정일 때 CLH가 완전히 기동하지 않은 상태에서 클라이언트가 접속을 시도하는 경우 비정상 종료하는 현상을 수정하였다.
- 관련 이슈 : [IMS 181211](#)

## 2.3.2. 클라이언트 / 서버

- **tmadmin API에서 tmax, tmaxnontx 게이트웨이 정보 조회 기능이 되지 않던 문제 수정**

- tmadmin API에서 tmax, tmaxnontx 게이트웨이 정보 조회 기능이 되지 않던 문제를 수정하였다.
- 관련 이슈 : [IMS 199091](#)

- **서버 프로그램에서 system(tmboot) 호출할 때 BOOT3007 오류 발생 현상**

- 서버 프로그램에서 system(tmboot)을 호출할 때 \$TMAXDIR/config/tmconfig 파일이 아닌 \$TMAXDIR/path/tmconfig 경로를 참조하도록 수정하였다.

- 관련 이슈 : [IMS 83216](#)
- **tpsleeper에서 이미 tpcall 응답이 들어왔음에도 깨어나지 않는 현상 수정**
  - tpsleeper에서 이미 수신한 tpcall, tprecv 메시지가 있는지 확인하도록 수정하였다. conversation 경우는 상대방측의 disconn 발생도 감지할 수 있도록 수정하였다.
  - 관련 이슈 : [IMS 83073](#)
- **tmadmin API를 하는 경우 메모리 누수 발생 수정**
  - tmadmin API에서 TMADM\_SVC\_STAT, TMADM\_SVC\_STAT\_EX를 조회할 때 메모리 누수가 발생하는 현상을 수정하였다.
  - 관련 이슈 : [IMS 103322](#)
- **MultipleRM를 사용하는 경우 tmax\_is\_xa() 함수 반환값 오류 수정**
  - MultipleRM를 사용하는 경우 tmax\_is\_xa() 함수가 XA임에도 Non\_XA가 반환되는 현상을 수정하였다.
  - 관련 이슈 : [IMS 109734](#)
- **tmadmin() API의 TMADM\_SVR\_CONF 명령 호출할 때 TPEINVAL 오류 발생 수정**
  - tmadmin() API의 TMADM\_SVR\_CONF 명령 호출할 때 TPEINVAL 오류가 발생하는 현상을 수정하였다.
  - 관련 이슈 : [IMS 114068](#)
- **tpcall 수행 후 타임아웃 발생 시 TPGOTSIG 에러가 출력되는 현상 수정**
  - TXTIME를 체크하여 만약 tx\_timeout이 발생했다면 TXETIME 에러가 출력되도록 수정하였다.
  - 관련 이슈 : [IMS 192335](#)
- **FDL 함수에서 LONG 타입을 사용할 때 비정상 동작하는 현상 수정**
  - 엔진은 64Bit, AP는 32Bit 환경에서 fbgetalloc\_tu, fbgetalloc\_tut 함수에서 LONG 타입을 사용할 때 비정상 종료하는 현상을 수정하였다.
  - 32bit 클라이언트에서 long에 -2를 fbput하고, 64bit 서버에서 LONG 타입에 fbget하면 양수가 출력되는 현상을 수정하였다.
    - 원인 : sign 부호를 저장하지 않고 (unsigned int)로 LONG 타입 변수에 값을 저장하기 때문에 음수든 양수든 모두 양수로 변환된다.
    - 해결 : sign 부호를 필드버퍼에 함께 저장하여 64bit 환경에서 fbget할 경우에 양수/음수 여부가 올바르게 변환되도록 수정하였다.
  - export TMAX\_FDL\_LONGSIGN=Y 를 지정해야 수정된 기능으로 동작한다.
  - 관련 이슈 : [IMS 86928](#)
- **RDP 프로그램에서 tp\_sleep()을 사용하였을 때 불특정 RDP 프로세스가 비정상 종료 현상 수정**
  - RDP 프로그램에서 tp\_sleep()을 사용하였을 때 불특정 RDP 프로세스가 비정상 종료되는 현상을 수정하였다.
  - 관련 이슈 : [IMS 17675](#)
- **서버를 재기동할 때에 TPCONNECT msg가 dicard되는 현상 수정**
  - 해당 메시지도 서버 큐에 저장하고 기동 이후 큐에서 꺼내 처리하도록 수정하였다.
  - 관련 이슈 : [IMS 99557](#)
- **tpstart 시에 main 서버로 연결이 되지 않을 때 재연결 타임아웃이 제대로 동작하지 않는 현상 수정**

- 시간 정밀도를 밀리세컨드 단위까지 정확하게 체크 하도록 수정하였다.
- 관련 이슈 : [IMS 99557](#)
- **TDL=Y, tx\_begin() 사용 시 서비스에서 tpforward(), tpreturn()으로 메시지의 길이가 10byte 늘어나는 현상 수정**
  - tpreturn(), tpforward()에서 암시적 정합성 제공을 위해 내부적으로 트레일러 메시지를 사용자 메시지 뒤에 덧붙여 Tmax 사이에 전달했는데, 트레일러 메시지의 길이를 잘못 계산하는 문제가 있다.
  - 트레일러 길이를 올바르게 계산하도록 수정하여, 사용자 메시지의 길이가 10byte 늘어나는 현상을 수정하였다.
  - 관련 이슈 : [IMS 114518](#)
- **tdlcall() 호출할 때 Core 발생 문제 수정**
  - Tmax 5에서 char[] 타입의 버퍼가 리팩토링으로 불필요해져 제거되어, Tmax 5 버전에는 Tmax 4 버전에 비해서 tdlcall() 함수의 stack frame 크기가 작다.
  - user\_function에서 stack overflow를 했을 때 Tmax 4 버전의 tdlcall()은 stack frame 크기가 커서 사용자 함수 호출 이후에 사용하는 return address, parameter가 깨지지 않는다. Tmax 5 버전은 위의 메모리가 깨지면서 Core가 발생한다.
  - Tmax 4 버전과 동일하게 사용자 함수에서 stack overflow를 발생시킬 때 tdlcall() 함수의 stack frame에 여유 버퍼(4000byte)를 추가해서 중요 메모리 영역을 보호하도록 수정하였다.
  - 관련 이슈 : [IMS 114518](#)
- **tpstart 없이 txinfo(NULL) 호출할 때 비정상 종료되는 현상 수정**
  - tpstart 없이 txinfo(NULL) 호출할 때 비정상 종료되는 현상을 수정하였다.
  - 관련 이슈 : [IMS 116481](#)
- **libclithr 라이브러리에서 컨텍스트 공유 시 tpreset() 호출 후 다른 스레드의 tpend에서 Core가 발생하는 현상 수정**
  - tpreset()를 실행할 때 여러 스레드가 컨텍스트를 공유한 상황이라면 TPEPROTO를 발생하도록 수정하였다.
  - 관련 이슈 : [IMS 138382](#)
- **tpcall(TPNOTIME) 호출후 대기중 signal 수신 시 TPEOS 발생 현상 수정**
  - TPSIGRSTRT flag가 설정되어 있으면 계속 응답을 기다리고, 설정이 없으면 TPGOTSIG 에러를 리턴하도록 수정하였다. tpcall, tpgetrply 등의 응답을 대기하는 함수에서 TPSIGRSTRT를 bitwise or 연산으로 지정하면 시그널을 수신하더라도 무시하고 다시 대기한다.
  - 관련 이슈 : [IMS 149444](#)
- **Tmax 서비스에서 버퍼 크기를 줄여 tprealloc() 호출 시 비정상 종료되는 현상 수정**
  - 크기가 줄어드는 경우에 부가정보(trailer)의 데이터를 임시 버퍼에 복사하여 메모리 접근 오류가 발생하지 않도록 수정하였다.
  - 만약 사용자가 설정된 버퍼 크기를 초과하여 데이터를 기록하는 경우(buffer overflow)에는 tprealloc()을 수행하는 경우 다음 에러를 출력하고 NULL을 리턴하도록 수정하였다.



SVR02300 The boundary of the buffer allocated by tmalloc() is corrupted: (buffer size)  
(trailer size) (boundary hex data)

◦ 관련 이슈 : [IMS 148854](#)

• **tpgetactivesvr API 사용 시 프로세스 메모리 증가 현상 수정**

◦ tpgetactivesvr API에서 두 번째 파라미터를 사용자 쪽에서 할당한 내용을 담게 해주었는데 해당 영역을 free하고 있지않아 memory leak이 발생하는 현상을 수정하였다.

◦ 관련 이슈 : [IMS 155493](#)

• **멀티 스레드 클라이언트에서 스레드가 Hang이 발생하는 오류 수정**

◦ 멀티 스레드 멀티 컨텍스트 환경에서 스레드 아이디가 4bytes 이상인 경우 그 이상 값이 잘린 채로 찾는 문제를 수정하였다.

◦ 관련 이슈 : [IMS 150129](#)

• **XAOPTION=DYNAMIC 환경에서 1PC 특정 시점에 불필요한 xa\_commit()을 호출하는 현상 수정**

◦ DB2를 사용하고 Tmax 환경설정 XAOPTION="DYNAMIC" 설정된 환경에서 tx\_commit()를 수행하는 경우 TX\_FAIL(-7) 에러가 발생하고, 다음과 같이 로그가 발생하는 현상을 수정하였다.

(E) SVR2037 internal\_commit\_error : XAER\_NOTA (the XID is not valid) [CSC5607]

◦ 관련 이슈 : [IMS 98123](#)

• **tpchkunsol()에서 블로킹되는 현상을 수정**

◦ tpchkunsol()에서 블로킹되는 현상을 수정하였다.

◦ 관련 이슈 : [IMS 120817](#)

• **tprecvfrom()에서 TPMSGDESC 인자 초기화 안하던 문제 수정**

◦ tprecvfrom()에서 TPMSGDESC 인자 초기화 안하던 문제를 수정하였다.

◦ 관련 이슈 : [IMS 127021](#)

• **멀티 스레드 클라이언트에서 TMAX\_TRACE 사용 시 Core 발생 현상 수정**

◦ 멀티 스레드 클라이언트에서 TMAX\_TRACE를 사용하는 경우 Core 발생하는 현상을 수정하였다.

◦ 관련 이슈 : [IMS 121250](#)

• **STD\_DYN 타입의 서버에 대한 tadmin(TMADM\_SPR\_STAT..) 조회 시 잘못된 spri 값을 전달해 주는 문제 수정**

◦ STD\_DYN 타입의 서버에 대한 tadmin(TMADM\_SPR\_STAT..)을 조회하는 경우 잘못된 spri 값을 전달해 주는 문제를 수정하였다.

◦ 관련 이슈 : [IMS 154342](#)

• **암호화 기능 사용 시 tpstart에서 TPEOTYPE, TPECLOSE 에러 발생 현상 수정**

◦ 암호화 기능을 사용하는 경우 tpstart에서 TPEOTYPE, TPECLOSE 에러가 발생하는 현상을 수정하였다.

◦ 관련 이슈 : [IMS 64262](#)

- **DYNSVR에서 요청한 서버에 tpsendto() 호출 시 에러 현상 및 메시지 수신 인자 잘못 전달 현상 수정**
  - DYNSVR에서 요청한 서버에 tpsendto() 호출 시 TPEINVAL 에러 현상 및 tprecvfrom에서 CLH로부터 메시지를 수신하는 경우 thrvar 인자 잘못 전달하는 현상을 수정하였다.
- **tprecvfromcli() 사용 시 요청한 상태에서 해당 노드가 연결이 종료되는 경우 무한대기를 타는 현상 수정**
  - 클라이언트로 tprecvfromcli() 요청 후 응답을 주기 전에 해당 클라이언트가 종료될 경우 요청자에게 tprecvfromcli() 함수에서 TPECLOSE 에러를 주도록 수정하였다.
  - tprecvfromcli() 호출할 때 클라이언트가 종료된 상태이면 clh에서 TPECLOSE 에러를 리턴하도록 수정하였다.
  - tprecvfromcli() 호출후 libsvr에서 응답을 수신하는 경우 seqno가 맞지 않으면 에러를 내도록 수정하였다.
  - tprecvfromcli()에 TPNOTIME flags 추가하였다.
  - tprecvfromcli()를 호출하는 경우 clid 값이 다른 노드의 클라이언트일 경우 해당 노드의 상태를 체크하여 노드가 다운된 상태일 경우 caller에 TPESVRDOWN 에러를 리턴하도록 수정하였다.
  - tpsendto(), tpsendtocli(), tprecvfromcli()에서 clid 값이 음수일 때 API에서 TPEINVAL 에러 리턴하도록 수정하였다.
  - 관련 이슈 : [IMS 193880](#)

### 2.3.3. 도메인 게이트웨이

- **CLH를 재기동할 때 lock file을 다른 프로세스가 사용 중인 경우 문제 수정**
  - CLH가 재기동할 때 도메인 게이트웨이의 -c 옵션이 설정되어 있으면(다른 서버 프로세스에서 path/clhld000 lock 파일을 사용 중) CLH가 기동과정에서 lockf() 함수에서 대기하는 문제가 있다.  
  
CLH가 기동하면 lock 파일에 의해서 대기상태에 빠지지 않도록, 기존 파일을 삭제하고 항상 새로운 파일을 생성하도록 수정하였다.
  - 관련 이슈 : [IMS 106794](#)
- **DomGW를 재기동할 때 발생하는 문제 수정**
  - DomGW를 재기동할 때 다음의 에러 메시지가 출력되는 문제를 수정하였다.

TMM3004 SVR (gw) is restarted the 1th time

  - 관련 이슈 : [IMS 115478](#)
- **CLOPT -p 옵션을 사용할 때 메시지 전문이 출력되지 않는 문제 수정**
  - CLOPT -p 옵션을 사용할 때 메시지 전문이 출력되지 않는 문제를 수정하였다.
  - 관련 이슈 : [IMS 114185](#)
- **RDOWN=Y 설정 시 리모트 도메인 tmdown 시 Hang 현상**
  - CLH는 서버 프로세스가 종료 될 때 해당 서버 큐에 쌓여있는 요청들이 BLOCKTIME이 지난 경우 요청자에게 에러 응답을 돌려주지 않고 있다. tmaxgw 및 tmaxgwnt 타입의 서버 프로세스가 종료될 때에는 CLH가 BLOCKTIME으로 검사하지 않도록 수정하였다.

tmaxgw 및 tmaxgwnt에서 RDOWN에 의해 종료할 때에 불필요하게 계속 CLH에게 종료 메시지를 보내는 부분이 수정되었으며, 이 과정에서 이미 닫힌 CLH와의 채널을 통해 메시지를 전송하려고 시도하던 부분 역시 수정되었다.

- 관련 이슈 : [IMS 114951](#)

## 2.3.4. TCP 게이트웨이

- **check\_end\_msg()에서 수신 데이터의 뒷 부분이 버려지는 오류 수정**

- check\_end\_msg()에서 수신 데이터보다 리턴한 데이터 크기가 작을 경우에 수신 데이터의 뒷 부분이 버려지는 오류를 수정하였다.
- 관련 이슈 : [IMS 99925](#)

- **TCPGW에서 set\_ping\_msg를 사용할 때 발생하는 문제 수정**

- set\_ping\_msg()가 init\_remote\_info()보다 먼저 호출되는 오류를 수정하였다.

TCPGW에서 set\_ping\_msg를 사용할 때 타임아웃 체크 주기가 잘못된 현상을 수정하였다. TCPGW 내부 체크 로직이 interval 주기로 ping msg를 보낸 후 interval 초 동안 타임아웃을 체크하지 않게 구현되어 있어 딜레이가 발생하였다.

ping 전송은 interval 주기로 보내고 전송 후 즉시 pong 기다리는 주기를 타임아웃 시간으로 동작하게 수정하였다.

- TCPGW에서 헬스체크 주기마다 메모리가 증가하는 현상을 수정하였다.

헬스 체크를 위해 ping 메시지를 리모트로 전송한 뒤 리모트에서 ping 응답(pong)이 수신되면 chk\_pong\_msg() 콜백함수가 호출된다. 이 콜백함수에서 양수를 리턴하면 헬스 체크용 메시지로 판단하고 get\_msg\_info() 콜백이 호출되지 않고 종료된다. 이때 pong 메시지에 대한 버퍼를 해제하지 않는 문제를 수정하였다.

- 관련 이슈 : [IMS 117622](#), [IMS 110764](#)

- **SVR9999 에러 메시지가 slog에 출력안되는 문제 수정**

- TCPGW에서 CLOPT -x 옵션에 의해 주기적으로 Ping 체크를 하는 데, 서비스 요청이 없는 상황에서 Ping 체크 타임아웃이 발생하는 경우 tmm으로 slog 메시지를 전송안하는 문제를 수정하였다.
- 관련 이슈 : [IMS 121367](#)

- **uid가 중복 발급되는 현상 수정**

- uid가 중복되지 못하도록 수정하였다.
- 지정된 범위 내의 모든 uid가 현재 사용 중이면, "uid is full" 메시지와 함께 -1을 리턴하고 종료하도록 수정하였다.
- 관련 이슈 : [IMS 126677](#)

## 2.3.5. Tuxedo 게이트웨이

- **TUXGW에서 log macro가 동작하지 않는 현상 수정**

- TUXGW에서 log macro가 동작하지 않는 현상을 수정하였다.

- 관련 이슈 : [IMS 108835](#)

- **alive check 응답 메시지가 에러 응답일 때 일반 메시지처럼 오인하여 에러 로깅되는 현상 수정**

- alive check 응답 메시지가 에러 응답일 때 일반 메시지처럼 오인하여 에러 로깅되는 현상을 수정하였다.

- 관련 오류 메시지

```
GATEWAY9003 Error reply from remote gateway :diagonostic [6] tran [0:0:0] [TUXGW0409]
```

- 관련 이슈 : [IMS 100700](#)

- **이미 에러 처리된 요청에 대해서 다시 정상 응답 메시지를 요청한 곳으로 반환하는 문제 수정**

- TUXGW에서 Tuxedo로의 요청에 대해서 타임아웃이 발생하면 해당 요청에 대해서 TPETIME 에러 응답을 요청한 곳으로 반환한다. 그런데 TUXGW로의 요청이 많은 부하 상황에서 Tuxedo로부터 타임아웃 처리된 응답이 들어오게 되면, 이미 에러 처리된 요청에 대해서 다시 정상 응답 메시지를 요청한 곳으로 반환하는 문제가 있다.

이 경우에 그 이후에 들어오는 요청은 다음의 에러가 발생하였다.

```
(W) GATEWAY0209 internal error : -1 [TUXGW0118]
```

TUXGW에서 이미 에러 처리한 요청에 대해서 Tuxedo로부터 응답이 들어올 경우에는 아래 로그를 출력하고 폐기시키도록 수정하였다.

```
(W) GATEWAY0209 internal error : invalid msg from rgw 0, maybe a message arrived after timeout, seqno 0 [TUXGW0161]
```

- 관련 이슈 : [IMS 101645](#)

- **tuxgw -n 옵션을 32bit Linux에서 사용할 때 TUXGW에서 비정상 종료 현상 수정**

- 관련 이슈 : [IMS 108634](#)

- **TUXGW alive check가 동작하지 않는 현상 수정**

- TUXGW Inbound XA 요청을 받은 후 응답을 주기 전에 alive check를 하면 이후 alive check를 하지 않는 현상을 수정하였다.

- 관련 이슈 : [IMS 108835](#)

- **TUXGW에서 프로토콜 에러가 발생하는 현상 수정**

- Tuxedo에서 null을 데이터로 반환했을 경우 TUXGW에서 프로토콜을 제대로 인식하도록 수정하였다.

```
(E) GATEWAY9002 Protocol error : [TUXGW0507]
```

- 관련 이슈 : [IMS 108835](#)

- **TUXGW Table overflow 에러 수정**

- tpcall을 호출하는 경우 내부 제한 값인 MAXSACALL 값이 설정되지 않는 오류를 수정하였다.

tpcall을 호출하는 경우 내부 제한 값인 MAXSACALL이 타임아웃이 발생하는 경우 테이블이 감소되지 않아 Table overflow 에러가 발생하였다. 이를 타임아웃될 때 테이블이 감소되도록 수정하였다.

- 관련 오류 메시지

```
(E) GATEWAY0209 internal error : request table overflow [TUXGW0014]
```

- 관련 이슈 : [IMS 109504](#)

- **TUXEDO와 연결이 종료되었을 경우 tuxgwa 비정상 종료 현상 수정**

- 관련 이슈 : [IMS 155400](#)

- **tuxgwa aix에서 도메인명 길이 잘못 해석하는 부분 수정**

- tuxgwa aix에서 도메인명 길이를 잘못 해석하는 부분을 수정하였다.

- 관련 이슈 : [IMS 155400](#), [IMS 156055](#)

- **tuxgw failover 및 connection 오동작 수정**

- clopt -n 옵션을 설정하는 경우 tuxgw main/backup 연결 이후 Tuxedo의 main domain 종료 이후 호출하는 경우 failover가 되지 않고 타임아웃이 발생하는 현상을 수정하였다.

- Tuxedo의 main/backup domain인 모두 종료된 상태에서 tuxgw를 기동하는 경우 GATEWAY0063 gethostbyname error 에러 발생하는 현상을 수정하였다.

- main/backup 설정 이후 backup 만 연결된 상태에서 Tuxedo backup domain 종료 이후 서비스를 호출할 때 타임아웃이 발생하는 현상을 수정하였다.

- Tuxedo가 재부팅 후 tmax에게 inbound 연결을 맺을 경우 Core를 내면서 서버가 재기동하는 현상을 수정하였다.

- 관련 이슈 : [IMS 105512](#)

## 2.3.6. Java 게이트웨이

- **JEUS type의 게이트웨이에서 alive check 옵션 사용 시 연결을 종료시키는 현상 수정**

- JEUS type의 게이트웨이에서 alive check 옵션 사용 시 in channel에 대해서도 alive check를 시도하여 연결을 종료시키는 현상 수정

- 관련 이슈 : [IMS 103171](#)

- **JEUSGW에서 MAXINRGW 비정상 동작 오류 수정**

- JEUSGW 인바운드를 연결하는 경우 MAXINRGW + 30 설정값만큼 연결되지 않는 현상을 수정하였다.

- 인바운드 세션에 대해 alive check하는 현상을 수정하였다.

- 관련 이슈 : [IMS 103171](#)

- **WebT에서 JEUSGW로 접속했을 때 RQ 서비스가 되지 않는 현상 수정**

- JEUSGW에서 tpstart를 수행하는 경우 전달해야 할 정보들을 관리하지 않고 있어 rq, node 정보를 가져올 수 있도록 JEUSGW, CLH, TMM의 동작을 수정하였다.

◦ 관련 이슈 : [IMS 104273](#)

• 게이트웨이에 잘못된 옵션을 설정한 경우 Core가 발생하는 현상 수정

◦ SUN 9/10에서 jeugw -a 등 잘못된 옵션을 설정하면 Core가 발생하여 수정하였다.

◦ 관련 이슈 : [IMS 114154](#)

• 게이트웨이에서 전문 로깅 기능을 사용할 때 게이트웨이가 비정상 종료하는 오류 수정

◦ JEUSGW, TMAXGW, TUXGW에서 아래와 같은 전문 로깅 기능을 사용할 때 %n 문자열이 포함된 경우 비정상 종료하는 오류를 수정하였다.

▪ JEUSGW

```
clopt="-D4"
```

▪ TMAXGW

```
clopt="-p"
```

▪ TUXGW

```
clopt="-D"
```

◦ 관련 이슈 : [IMS 115860](#)

• SUN에서 JEUSGWA 아웃바운드를 요청하는 경우 Core 발생 현상 수정

◦ 관련 이슈 : [IMS 105859](#)

• JEUSGW, JEUSGW Async에서 리모트와 연결이 되어도 로그가 출력 안되는 현상 수정

◦ 연결에 실패하는 경우 다음 로그들이 출력된다.

▪ 리모트로부터의 연결 요청에 대한 수락 실패

```
(E) GATEWAY3010 connection error from remote gateway [JGW0002]
```

▪ 리모트 주소가 1개일 때

1) 즉시 에러 발생

```
(I) GATEWAY0046 socket connect error :rgw(c) host[xxx.xxx.xxx.xxx], port[portno],  
type[16(primary)|32(backup)],  
[재시도 index : 신규연결채널 index] [COM3605]
```

2) 요청후 대기 중 에러 발생

```
(I) GATEWAY0046 socket connect error :rgw(c) host[xxx.xxx.xxx.xxx], port[portno],
```

```
type[16(primary)|32(backup)],  
[0:0] [COM3609]
```

- 리모트 주소가 2개 이상일 때

#### 1) Primary 연결 요청이 실패

```
(I) GATEWAY0046 socket connect error : xxx.xxx.xxx.xxx portno [JGW0109]
```

#### 2) Backup 연결 요청이 실패

```
(I) GATEWAY0046 socket connect error : xxx.xxx.xxx.xxx portno errno [JGW0112]
```

- 관련 이슈 : [IMS 111280](#)

#### • JEUSGW CLOPT -D2 또는 -D3 옵션 설정 시 비정상 종료되는 현상 수정

- JEUSGW CLOPT -D2 또는 -D3 옵션을 설정하는 경우 비정상 종료되는 현상을 수정하였다.
- 관련 이슈 : [IMS 127689](#)

#### • JEUSGW에서 CLH로부터 받은 tpconnect reply 처리 이후 rgwi를 해제해서 그 이후 tpreturn에 대한 응답을 못하는 현상 수정

- JEUSGW에서 CLH로부터 받은 tpconnect reply 처리 이후 rgwi를 해제해서 그 이후 tpreturn에 대한 응답을 못하는 현상을 수정하였다.
- 관련 이슈 : [IMS 152185](#)

## 2.3.7. Entera 게이트웨이

#### • Entgw memory leak 수정

- Entera server로부터 요청을 받아서 서비스를 수행하고 응답을 줄 때 일부 데이터를 free하지 않는 문제를 수정하였다.
- 관련 이슈 : [IMS 101036](#)

#### • entgw XA가 되지 현상을 수정

- entgw XA가 되지 현상을 수정하였다.
- 관련 이슈 : [IMS 101036](#)

## 2.3.8. TDL

#### • SunOS에서 tdlupdate 시 static 함수를 대상에 포함시키는 문제 해결

- SunOS에서 tdlupdate()를 수행하는 경우 static 함수를 대상에 포함시키는 문제를 해결하였다.
- 관련 이슈 : [IMS 98172](#)

- **tdlupdate -l -m 사용 시 파일 목록에서 문제가 생기면 진행이 중단되는 현상 수정**

- 리스트에 있는 파일을 검사하여 이상이 없는지 여부를 확인하고 업데이트를 진행하고 문제가 있다면 업데이트를 하지 않도록 수정하였다.
- TMAX\_ERR\_MSG, TMAX\_DEBUG 설정으로 tdlupdate 결과를 로그 파일로 남길수 있도록 수정하였다.
- 관련 이슈 : [IMS 104752](#)

- **AIX에서 TDL 공유 메모리(tdlshm)에 변수가 함수처럼 올라가는 현상 수정**

- AIX 플랫폼의 특성상 nm으로 전역 변수까지 검색되어 변수가 TDL 함수로 등록되는 현상을 수정하였다.
- 관련 이슈 : [IMS 107466](#)

- **tdlupdate -p 업데이트할 때 tdlupdate 오류 발생하는 현상 수정**

- tdlupdate -p 업데이트할 때 tdl.cfg(LANG=COBOL) 설정이 있을 경우 tdlupdate 오류가 발생하는 현상을 수정하였다.
- 관련 오류 메시지

```
(E) TDLUTIL0078 file rename error: /run/tdlcall2v_s.so -> /run/tdlcall2v_s.5728054d [TDL0447]
[No such file or directory]
```

- 관련 이슈 : [IMS 110591](#)

- **tdlupdate가 수행되는 동안 tdlcall()이 호출되면 에러가 발생하는 현상 수정**

- tdlupdate가 수행되는 동안 tdlcall()이 호출되면 "library not found" 에러가 발생하는 현상을 수정하였다.
- tdl.cfg의 MODLE=SINGLE 구성에서만 발생한다.
- 관련 이슈 : [IMS 113724](#)

- **tdlupdate -l list 수행할 때 list 파일이 null인 경우 오류가 발생하는 현상 수정**

- 관련 오류 메시지

```
(I) TDLUTIL0040 invalid arguments [TDL0401]
(E) TDLUTIL0001 memory allocation error : size = 0 [TDL0434]
```

- 관련 이슈 : [IMS 114469](#)

## 2.3.9. 유틸리티

- **tmapm이 RUN 상태인 TMAXNONTX 타입 게이트웨이를 체크하는 현상**

- tmapm이 TCS, UCS 타입의 서버에만 체크하도록 수정하였다.
- 관련 이슈 : [IMS 96726](#)

- **ENVFILE을 사용할 때 큰따옴표(" ") 안에 내용이 인식하도록 수정**

- ENVFILE에 지정하는 환경변수 파일에서 환경변수 값에 등호(=) 문자가 들어있을 경우 에러가 발생하였다.



◦ 관련 이슈 : [IMS 97888](#)

• **CFL3110 출력 오류 수정**

◦ CFL3110을 출력할 때 실제 CPC 값 보다 크게 나오는 현상을 수정하였다.

```
(E) CFL3110 more CPC(100) than MAXCPC(3) are defined at node [CFL0955]
```

◦ 관련 이슈 : [IMS 102921](#)

• **STD 타입/ DYN 타입의 서버 SPR 개수를 32768 ( 65536 / 2 ) 보다 크게 설정한 경우 오류가 발생하는 현상**

◦ 서버의 타입에 따라 MIN / MAX의 값에 따라 최대 spr 갯수를 합산해서 그것이 65536의 절반인 32768을 넘을 경우 cfi에서 에러를 출력하도록 수정하였다. STD의 경우 spr에 TMS가 포함된다.

◦ 관련 이슈 : [IMS 103458](#)

• **mksvr 사용 시 svctab 권한이 없을 경우 비정상 종료하는 현상 수정**

◦ 해당 사용자가 파일 읽기 권한이 없을 경우 아래의 에러 메시지를 출력한다.

```
cannot open [파일명] : Permission denied
```

◦ 관련 이슈 : [IMS 118406](#)

• **racd에 의해 기동된 tadmin이 exit() 함수에서 무한루프에 빠지면서 종료되지 않는 현상 수정**

◦ tadmin에서 종료 진행 중에 signal을 수신할 때 재귀적으로 exit()가 호출되지 않도록 수정하였다.

◦ 관련 이슈 : [IMS 129272](#)

• **cfi에서 cfgadd로 서버 서비스를 추가할 때 메모리의 상황에 따라 실패하는 현상**

◦ 메모리 오류로 추가된 내역의 순서가 바뀌면서 TMM에서 에러가 발생하지 않도록 수정하였다.

◦ 관련 이슈 : [IMS 111598](#)

• **tdlinit을 하지 않고 tmboot 시 CLH가 죽는 현상 수정**

◦ tdlinit을 하지 않고 tmboot를 수행할 때 tdlinit을 하라는 경고 메시지를 띄워준 후 tmboot 종료하도록 수정하였다.

◦ 관련 이슈 : [IMS 116693](#)

• **tmdown -b -g 또는 tmdown -b -A 수행시점에 NRDY 상태의 서버에 대해서는 POD 서버에서 boot block 기능이 동작하지 않는 현상 수정**

◦ tmdown -b -S 옵션은 tmdown에서 tmm으로 요청하지 않는 문제 수정하였다.

◦ tmdowb -b -g, -A 옵션은 tmm에서 모든 서버들에 대해서 boot block을 적용하도록 수정하였다.

◦ 관련 이슈 : [IMS 174881](#)

• **User License 사용 시 tmboot에서 비정상 종료 발생 현상 수정**

◦ User License를 사용하는 경우 tmboot에서 Core 발생 현상을 수정하였다.

◦ 관련 이슈 : [IMS 169168](#)

• **Linux 64bit 환경에서 32bit 바이너리 적용 후 mksvr(mkcli) 실행 시 make server(client) fail 발생 현상**

## 수정

- linux에서 -64, -32 옵션에 따라 CFLAGS에 -m64, -m32를 붙이도록 수정하였다. -u 옵션을 사용하는 경우 적용하지 않는다.
  - 관련 이슈 : [IMS 169168](#)
  - **tmaxlibver 수행 시 해당 라이브러리의 undefined symbol이 있으면 버전 확인이 불가능한 현상 수정**
    - tmaxlibver를 수행하는 경우 해당 라이브러리의 undefined symbol이 있으면 버전 확인이 불가능한 현상을 수정하였다.
    - 관련 이슈 : [IMS 99925](#)
  - **cfl 비밀번호 입력 시 notty 환경에서 입력 실패하는 현상 수정**
    - crontab 등 notty 환경에서 cfl을 수행할 때 stdin으로부터의 입력을 제대로 처리하지 못하는 문제로 인해 아래와 같은 에러가 출력되는 현상을 수정하였다.

```
(E) CFL0247 wrapped cipher text is too long : 5 > 256 [COM0706]
```

  - 관련 이슈 : [IMS 149970](#), [IMS 149156](#)
- **racd에서 수행한 cfl 등의 명령이 20초 이상 수행되는 경우 요청한 노드로 강제로 응답을 주는 현상 수정**
  - racd에서 tmdown, tmboot의 명령어 실행한 뒤 종료시까지 대기하는 시간을 20초에서 기본 60초로 변경하고, cfl, tdl의 요청은 완료할 때까지 무제한 대기하도록 수정하였다.
  - racd의 tmdown, tmboot의 기본 대기 시간에 대해서 옵션 -t sec를 추가하였다. (음수 지정하는 경우 무제한 대기, 1 이상의 값 지정하는 경우 해당 시간만큼만 대기, 0 지정 불가)
  - racd에서 명령 실행 중에 타임아웃이나 에러가 발생하면 cfl, tmboot, tmdown, racdr 유틸리티에서 이와 관련된 내용을 출력하도록 수정하였다.
  - cfl에서 SIGPIPE가 발생하더라도 이를 무시하도록 수정하였다.
  - 관련 이슈 : [IMS 176734](#)
- **멀티 노드에서 OPENINFO절 부분암호화 설정 후 cfl 시 9자리 이상 입력받을 경우 CFL3152 에러 현상 수정**
  - 멀티 노드 환경에서 암호화된 password 값을 리모트에서 읽을 때 whitespace를 잘못 파싱하는 문제 수정하였다.
  - 서버 그룹이 2개 이상일 경우 에러가 나는 현상 수정하였다.
  - 한 OPENINFO 절에 \*\*\*\*\*이 여러 개일 때도 rcfl이 동작하도록 수정하였다.
  - 다만 암호화 미입력 후 엔터 쳤을 때 rcfl이 동작하는 않는 현상은 수정하지 못하였다.
  - 관련 이슈 : [IMS 180449](#)

## 2.3.10. 관리 도구

- **st -s/p -x gateway\_name 잘못 표기되는 문제 수정**
  - st -s/p -x gateway\_name의 mintime, utime\*, stime\*이 잘못 표기되는 문제 수정하였다.
  - CPU Time을 사용하지 않는 게이트웨이 타입의 경우 usertime, systime을 사용하지 않도록 수정하였다.

(0에서 변경되지 않음)

- 관련 이슈 : [IMS 103525](#)
- **tmadmin st -t -x 수행 시 TMS 서버가 없으면 ADM0003 에러가 발생하는 현상 수정**
  - 해당 에러를 출력하지 않도록 수정하였다.
  - 관련 이슈 : [IMS 175706](#)
- **tmadmin에서 gwi 모니터링할 때 No Alive Connection 나오는 현상 수정**
  - alive check 메시지를 전송할 때 해당 세션의 상태를 READY에서 PING\_REQUESTED로 변경하고, alive check 응답이 들어오면 다시 READY로 변경한다. 이 사이에 tmadmin gwi 명령이 수행되면, 게이트웨이(Tmax/JEUS/Tuxedo)에서 상태 정보를 체크하게 되는데 이때 READY 상태의 세션만 체크하고 있어서 No Alive Connection으로 나타나는 현상을 수정하였다.
  - JEUSGW에서 XA 처리 상태에서도 No Alive Connection 발생하고 있어 이에 대해서도 발생하지 않도록 수정하였다.
  - 관련 이슈 : [IMS 112228](#)

## 2.3.11. HostLink

- **LU62R POSTSVC로 유발된 중복 로깅 에러 수정**
  - LU62R POSTSVC로 유발된 중복 로깅 에러를 수정하였다.
  - 관련 이슈 : [IMS 34331](#), [IMS 98180](#)
- **타임아웃으로 인한 에러 응답 시 사용자 헤더 덮어쓰는 현상을 수정**
  - 타임아웃으로 인한 에러가 응답되는 경우 사용자 헤더 덮어쓰는 현상을 수정하였다.
  - 관련 이슈 : [IMS 34043](#), [IMS 98180](#)
- **lu62sgw 에러 발생 시 에러코드를 잘못된 값으로 셋팅하는 현상**
  - primary\_rc가 셋팅되어 있으면 Tmax 에러코드가 아닌 해당 값을 리턴하도록 되어 있는데, 항상 Tmax 서비스로 응답이 전달되어야 하므로 primary\_rc에 값이 셋팅되지 않도록 수정하였다.
  - 관련 이슈 : [IMS 79965](#)
- **lu62rgw에서 DTP 설정 시 2번째 수신할 때 비정상 종료되는 현상**
  - lu62rgw에서 DTP 설정시 2번째 수신할 때 비정상 종료되는 현상을 수정하였다.
  - lu62rgw에서 데이터 수신중 AP\_DATA\_INCOMPLETE 발생했을 때 기존 수신했던 데이터를 잃어버리는 현상을 수정하였다.
  - 관련 이슈 : [IMS 100643](#)
- **lu62r sysplex 도입에 따른 FQPLUNAME 동작의 차이**
  - hkcf에서 lu62rgw에 한해서, FQPLUNAME 항목을 다음과 같이 설정할 수 있도록 기능을 확장하였다.
    1. FQPLUNAME 항목을 설정하지 않아도 hkcf이 가능하도록 수정하였다(기존에는 반드시 지정해야 하는 필수 옵션이었음).

설정하지 않을 경우에는 communication server에 LU alias에 대응되는 모든 PLUNAME으로부터의 요청들을 처리할 수 있게 된다.

2. FQPLUNAME 항목에 구분자 콤마(,)를 통해 하나 이상의 pluname을 지정할 수 있도록 하였다(최대 32개까지 지정할 수 있음).

구분자 외에 공백이 들어가서는 안된다.

예)

```
FQPLUNAME="GBOK01.S2T0,GBOK01.S0T0"
```

- liblu62rgw 라이브러리에서는 구분자 콤마로 여러 개의 FQPLUNAME 이 지정된 경우 각각의 FQPLUNAME 모두에 대해서 communication server에 attach notify를 받을 수 있도록 적용하였다.
- HostLink의 공유 메모리(shared memory) 구조가 변경되었다.

- 관련 이슈 : [IMS 101080](#)

- **lu62rgw 부하상황에서 비정상 종료 현상 수정**

- lu62rgw 부하상황에서 비정상 종료 현상을 수정하였다.
- 관련 이슈 : [IMS 101112](#)

- **HOSTLINK 환경설정 파일 설정 후 메모리가 증가하는 현상 수정**

- HOSTLINK 환경설정 파일 LOGGING 절을 설정하고 운영하면, 데이터를 로깅할 때마다 메모리가 증가하는 현상을 수정하였다.
- 관련 이슈 : [IMS 102206](#)

- **LU62RGW가 비정상 종료하는 현상 수정**

- LU62RGW에서 메인프레임으로부터 SNA를 통해 다수의 트랜잭션 요청이 동시에 들어오는 상황에서 LU62RGW 프로세스가 stack overflow가 발생하면서 비정상 종료하는 현상을 수정하였다.
- 관련 이슈 : [IMS 102206](#)

## 2.3.12. RPC

- **RPC 컴파일할 때 오류 발생 수정**

- RPC 컴파일할 때 "Too many arguments specified for macro dce\_dbgwrite" 오류가 발생하여 tmaxrpc.h를 수정하였다.
- 관련 이슈 : [IMS 82010](#)

- **blob에 대해서 select가 되지 않고 비정상 종료하는 현상 수정**

- Entera 서버로부터 요청을 받아서 서비스를 수행하고 응답을 줄 때 일부 데이터를 free하지 않는 문제를 수정하였다.
- 관련 이슈 : [IMS 101036](#)

- **tmaxrpc idlc에서 delphi xe 지원 시 char[][] 형을 인자로 받는 함수의 타입을 수정**

- tmaxrpc idlc에서 delphi xe를 지원하는 경우 char[][] 형을 인자로 받는 함수의 타입을 수정하였다.
- 관련 이슈 : [IMS 131289](#)

- **tmaxrpc server에서 SQL insert 수행 시 에러가 발생해야 하는 상황에서 정상처리되는 현상 수정**
  - Dynamic SQL 사용하는 경우 update/delete/insert는 EXEC SQL EXECUTE 로 처리하도록 수정하였다.
  - ttpc로 tbr.tbc 프리컴파일을 수행하는 경우 INSERT\_NO\_DATA\_ERROR=YES 설정을 추가하여 insert를 수행할 때 sqlca.sqlerrd[2]의 반영된 row 개수가 0인 상황에서는 에러가 발생하도록 수정하였다.
  - 환경변수 TMAX\_PRB\_RPC\_NODATA\_ERROR=Y 옵션을 지정할 때 update/delete/insert 상황에서 sqlca.sqlerrd[2] row 개수가 0인 경우 강제 에러처리 하는 기능 추가하였다.
  - Tiberco RPC 라이브러리에서 userlog 로그 내용에 ORACLE로 표기되는 부분 수정 및 에러코드를 출력하도록 수정하였다.
  - 관련 이슈 : [IMS 162408](#)
- **tpgetpeername이 IPv6 관련 패치에 따른 rpc 모듈 수정**
  - tpgetpeername이 IPv6 관련 패치됐는데 rpc 모듈에서 그에 맞게 변경이 안된 부분을 수정하였다.
  - 관련 이슈 : [IMS 170239](#)
- **TMAX\_TRACE 설정 시 tmaxrpc에서 제공하는 API들도 로그를 남기도록 수정**
  - TMAX\_TRACE를 설정하는 경우 tmaxrpc에서 제공하는 API들도 로그를 남기도록 수정하였다.
  - 관련 이슈 : [IMS 170072](#)
- **BLOB 및 LongRaw 타입 조회 시 길이가 0(null)이면 더이상 데이터 처리를 진행하지 않고 해당 요청을 에러로 리턴하는 현상 수정**
  - TMAX\_PRB\_RPC\_BLOB\_NULL=Y 환경변수를 지정하면 null의 경우에도 서비스를 정상 처리하도록 수정하였다.
  - 관련 이슈 : [IMS 173317](#)

## 2.3.13. Tcache

- **pfmTCacheInvalidateAllItems 내부에서 tpinit 시 실패하는 문제 수정**
  - 클라이언트에서 직접 tpstart나 tpinit을 하지 않았다면 pfmTCacheInvalidateAllItems가 실패로 처리되는 문제를 수정하였다.
  - 관련 이슈 : [IMS 106009](#)
- **tcacheadmin -u에서 free\_list entry unlock하지 않는 문제 수정**
  - tcacheadmin -u에서 실제 데이터 entry에 대한 lock 만 해제해 주고 free\_list entry를 unlock하지 않는 부분을 추가로 unlock하도록 수정하였다.
  - TCache 관련 실행 파일과 라이브러리에서 DEBUG log가 출력 가능하도록 추가하였다. TMAX\_DEBUG, TMAX\_LOGLVL 환경변수를 선언해야 한다.
  - 관련 이슈 : [IMS 110766](#)

## 2.3.14. 기타

- **libtmaxgwmt.so에서 slog로 메시지를 보내지 않는 현상 수정**

- libtmaxgw 계열의 라이브러리에서 에러 메시지를 출력하는 경우 ulog에만 기록되고 slog에 기록되지 않는 현상을 수정하였다.
- 관련 이슈 : [IMS 152540](#)
- **libtmaxgw에서 argument 분석 중에 에러 출력 이후 비정상 종료하는 현상 수정**
  - libtmaxgw에서 argument 분석 중에 에러 출력 이후 비정상 종료하는 현상을 수정하였다.
  - 관련 이슈 : [IMS 178376](#)

## 3. Tmax 6

본 장에서는 Tmax 6 릴리즈에서 추가된 새로운 기능과 변경된 기능에 대해 간략히 설명한다.

### 3.1. 신규 기능

#### 3.1.1. Tmax Deployment 기능 추가

서버 애플리케이션, 리소스 파일, 환경설정 파일 등을 배포하고 기동하는 것을 관리하는 프레임워크가 도입으로 이력과 버전들을 관리 기능 제공한다.

#### 3.1.2. TmaxGrid 기능 추가

TmaxGrid는 멀티 노드 환경에서 in-memory 데이터를 동기화시키는데 목적두며, 다수의 업무 프 로그램에서 이 기능을 이용해서 분산 lock, queue 기능, 공유 설정을 별도의 구현 없이 쉬운 API를 통해서 멀티 노드에서 구현 기능을 제공 한다. 자세한 사용법은 "TmaxGrid User Guide"를 참고한다.

#### 3.1.3. Multiple RM에 DB2 지원

Tmax 6부터 Multiple RM에 DB2를 지원한다. 자세한 사용법은 "Tmax Application Development Guide"를 참고한다.

#### 3.1.4. 보안 기능 강화 및 유연성 제공

기존 서버간의 암호호화를 할 때 Tmax에서 제공하는 기본 보안 모듈만 사용 가능하였으나, 사용자가 직접 원하는 보안 모듈을 적용할 수 있도록 보안 기능의 유연성을 제공한다. 이전 버전에서는 암호화를 CLH에서 담당 하였으나 이를 분리 CAS에서 수행하도록 분리해서 스레드를 제공하여 암호화 성능이 향상되었다.

#### 3.1.5. PHP 인터페이스 제공

PHP에서 Tmax 클라이언트의 기능을 사용할 수 있는 확장 모듈을 제공한다. 자세한 사항은 "Tmax Programming Guide (4GL)"를 확인한다.

### 3.2. 기능 개선

#### 3.2.1. 이름 길이 확장

기존에 설정 값을 15자리까지 사용하던 이름의 길이를 63자리까지 사용할 수 있도록 확장하였다.

- 세부 설정 값

```
Domain Name | Node Name |SVRGROUP |Server Name |Service Name |OWNER |HMSNAME |Gateway Name |RQ
Name(62 자리)
```

### 3.2.2. 시간 정밀도 개선

이전 버전의 시간 정밀도는 초 단위였으나 Tmax 6 이후로는 시간 정밀도 밀리세컨드 단위로 변경하였다. SVCTIME, TXTIME, BLOCKTIME 등 환경 설정 관련한 시간 정밀도는 밀리세컨드 단위로 입력을 받는다.

- 입력 방식
  - Tmax v5.0 이전의 방식과 호환되며, sec.millisecond로 설정한다. 예를 들면 10.001, 10이 가능하다. '10.'처럼 소수점 뒤에 숫자가 없으면 CFL 에러가 발생하므로 주의한다.
  - slog, ulog, debug log, svclog 등 로그 관련한 시간 정밀도는 마이크로세컨드 단위로 출력한다.
- 출력 방식
  - Tmax v5.0 이전의 방식에서 HHMMSS뒤에 소수점과 마이크로세컨드가 붙는다.

예)

```
svr333.1005.143119.511001
```

- User API가 추가되었다.

```
int tpsetsvctimeout(int msec, long flags)
int tx_set_transaction_mtimeout(TRANSACTION_TIMEOUT timeout)
```

- tadmin의 repeat(r) 명령어의 -i s(초) 옵션의 s가 밀리초로 변경되었다.

### 3.2.3. tmboot / tmdown 로깅 메시지 개선

- tmboot / tmdown 로깅 메시지 개선

tmboot 또는 tmdown할 때 slog에 아래와 같이 메시지가 출력되도록 기능이 개선되었다(%2 부분에 입력한 인자이다).

```
TMM01128 TMB00T registered, pid = %1 (%2)
TMM01129 TMDOWN registered, pid = %2 (%2)
```

예)

```
TMM.3721.112533.526303:(I) TMAX01128 TMB00T registered, pid = 3733 (tmboot -S svr2 ) [TMM0210]
```



```
TMM.3721.112533.536627:(I) TMAX01128 TMB00T registered, pid = 3736 (tmboot -n tmax8 -S svr1 )
[TMM0210]
```

- 관련 이슈 : [IMS 38782](#)

### 3.2.4. Tmax Log 메시지 통합

여러 메시지의 가독성 및 용이한 분석을 위해 기존 모든 메시지들의 prefix가 "모듈4자리번호"에서 "TMAX5자리번호"로 변경되었다. 또한 모듈에 관계 없이 'TMAX'를 접두어로 사용한다.

예)

```
SVR2058 -> TMAX02058
CLH2058 -> TMAX02058
```

### 3.2.5. Debug/Release Binary 통합

기존 디버깅 바이너리를 일반 모듈로 통합하였다. 기존에는 디버깅 바이너리를 적용을 위해서는 디버깅 바이너리 교체 후 엔진을 재기동 해야만 했으나 동적 LOGLEVEL 변경을 통해 실시간으로 디버깅 적용 가능하게 변경하였다.

tmm에 접속하지 않는 일반 모듈(cfl/uncfl/client 등)은 TMAX\_ERR\_MSG/TMAX\_DEBUG/TMAX\_LOGLVL의 설정을 해야 한다. TMAX\_ERR\_MSG=Y일 경우 TMAX\_LOGLVL=[DEBUG1~DEBUG4] 레벨에 디버그 로그 생성 로그는 TMAX\_DEBUG 설정에 따라 로그 파일이 생성된다.

- tmm에 접속하는 모듈은 Tmax 환경 파일에 설정한 각 모듈별 디버그 로그를 설정해야 한다.

```
TMMLOGLVL CLHLOGLVL TMSLOGLVL LOGLVL GATEWAYLOGLVL 별
```

내부적인 디버그 로그의 상세 정도를 DEBUG1, DEBUG2, DEBUG3, DEBUG4로 표기한다. (DEBUG4 가 가장 상세함)

- DEBUG1 : 일반적인 레벨
- DEBUG2 : 트랜잭션 레벨
- DEBUG3 : 그 외(주로 네트워크 관련)의 레벨
- DEBUG4 : 스레드 레벨

### 3.2.6. clilib, svrlib에 tmaxoserror 함수 추가

tmaxoserrno 변수를 통해서 TPEOS, TPESYSTEM이 발생하는 경우 OS syscall을 수행할 때 errno에 대해서 참조할 수 있도록 한다.

### 3.2.7. Svcscrpt 사용자 로깅 지원

- SVCLOG\_FORMAT 기능추가

svclog 의 항목을 사용자가 조정할 수 있도록 기능 추가 기존에 출력되던 항목은 변경할 수 없으며 이번에 추가하는 필드만 변경 가능하다. tadmin api, tip, snmp, webagent에서 추가된 SVCLOG\_FORMAT은 참조할 수 없다.

### 3.2.8. 도메인 게이트 웨이의 DIRECTION 동적 변경 기능 추가

- tadmin의 set 명령어를 통해서 게이트웨이의 direction 항목을 변경 가능하도록 기능이 추가되었다.

```
echo set -w gw3 direction 'BIDIR|B | IN|I | OUT|O' | tadmin -m
```

- 관련 이슈 : [IMS 43965](#)

### 3.2.9. 서버 프로세스들에 대한 기동시각 출력기능 추가

- 서버 프로세스들에 대한 기동시각 출력기능이 추가되었다.
  - tadmin api의 아래 구조체에 boot\_time 필드 추가(unix time)

```
- struct tmadm_spr_stat_body
- struct tmadm_spr_stat_ex_body
- struct tmadm_svr_stat_body
  struct {
    int sec;    // unix time second
    int msec;  // millisecond
  } boot_time;
```

- TIP server 에 필드키 추가(unix time)

```
- TIP_BOOTTIME_SEC
- TIP_BOOTTIME_MSEC
```

- tadmin의 st -p/st -v에 -b 옵션 추가(local time)

- 관련 이슈 : [IMS 78473](#)

### 3.2.10. MSGSIZEWARN, MSGSIZEMAX 옵션 추가

이전 버전에서는 CLH는 1GB가 넘는 단일 메시지가 들어오면 해당 client와의 연결을 종료시키면서 연결을 허용하지 않고 있으며, 이 기능을 확장하여 메시지의 최대 크기를 사용자가 지정할 수 있도록 하는 옵션이 추가되었다.

- MSGSIZEWARN, MSGSIZEMAX 옵션 추가

- MSGSIZEWARN

클라이언트/서버로부터 수신하려는 메시지의 크기(순수한 사용자 데이터 크기) 또는 talloc을 통한 버퍼 할당 크기가 지정한 값을 초과하면 SLOG에 warning 메시지를 출력한다. 메시지는 정상적으로 수신하고 처리한다.

- 단위 : Byte
- 기본값 : 1,073,741,824(1GB)
- 지정범위 : 1,024 ~ 1,073,741,824(1GB)

```
(W) TMAX02066 client(192.168.51.185) message size(5001) exceeds MSGSIZEWARN(2000),
service(TOUPPER) : 3 5013 5001 1342177292 [COM3808]
(W) TMAX02066 server(svr2:66) message size(5001) exceeds MSGSIZEWARN(2000),
service(TOUPPER) : 1003 5023 5001 1342177302 [COM3808]
```

- MSGSIZEMAX

클라이언트, 서버로부터 수신하려는 메시지의 크기(순수한 사용자 데이터 크기) 또는 talloc을 통한 버퍼 할당 크기가 지정한 값을 초과하면 SLOG에 error 메시지를 출력한다. 메시지를 무시하고, 해당 채널의 연결을 종료시킨다.

- 단위 : Byte
- 기본값 : 1,073,741,824(1GB)
- 지정범위 : 1,024 ~ 1,073,741,824(1GB)

```
(E) TMAX02065 client(192.168.51.185) message size(10001) exceeds MSGSIZEMAX(10000) : 3
10013 10001 1879048204 [COM3807]
```

- 관련 모듈

- CLH : 클라이언트, 서버로부터 들어오는 메시지
- jeusgw, tmaxgw, tmaxgwnt : 리모트 게이트웨이로부터 들어오는 메시지
- 다음의 명령으로 MSGSIZEMAX 변경한다.

```
set -n <nodename> msx <data size>
```

- 관련 이슈 : [IMS 38102](#)

### 3.2.11. runtime 설정 강화

- tmaxadmin 동적 변경 범위 확대
  - 타임아웃과 관련된 항목

- CHKINT(체크 인터벌) 관련 항목
- 로그 레벨
- 용이한 환경 파일 추가를 위해 cfl -a 방식이 변경되었다. 기존 제약 사항은 유지한다. (기존 항목 삭제 불가)
  - 변경전

```
cfl -i ori.m -a add.m -o tmchg
```

- 변경 후

```
cfl -a modify.m -o tmchg
```

### 3.2.12. runtime 로그레벨 변경 강화

tmadmin의 chlog2 명령어를 이용하여 런타임 도중에 로그 레벨을 동적으로 변경할 수 있는 기능이 추가되었다.

변경 가능한 타입은 TMM, CLH, TLM, CAS, RS, SQ, TG, TMS, HMS, RQ, GW, SVG, SVR, CLL 이다.

```
$$1 tmaxs1 (tmadm) : chlog2 -t [type_name] [-n name] -l [loglvl]
```