

# ENPHAROS TP

# Installation Guide

ENPHAROS v5.0



**DabomSoft**  
Pharos for one, Pharos for all

Copyright © 2014 Dabomsoft Co.,Ltd. All rights Reserved.

## Copyright Notice

Copyright © 2014 Dabomsoft Co., Ltd. All Rights Reserved.

대한민국 서울시 구로구 디지털로 306, 714호(구로동, 대륭포스트2차)

## Restricted Rights Legend

All Dabomsoft (Dabomsoft®) and documents are protected by copyright laws and the Protection Act of Computer Programs, and international convention. Dabomsoft software and documents are made available under the terms of the Dabomsoft License Agreement and may only be used or copied in accordance with the terms of this agreement. No part of this document may be transmitted, copied, deployed, or reproduced in any form or by any means, electronic, mechanical, or optical, without the prior written consent of Dabomsoft Co., Ltd.

이 소프트웨어(Dabomsoft®) 사용설명서의 내용과 프로그램은 저작권법, 컴퓨터프로그램보호법 및 국제 조약에 의해서 보호받고 있습니다. 사용설명서의 내용과 여기에 설명된 프로그램은 Dabomsoft Co., Ltd.와의 사용권 계약 하에서만 사용이 가능하며, 사용권 계약을 준수하는 경우에만 사용 또는 복제할 수 있습니다. 이 사용설명서의 전부 또는 일부분을 Dabomsoft의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단을 사용하여 전송, 복제, 배포, 2차적 저작물작성 등의 행위를 하여서는 안 됩니다.

## Trademarks

ENPharos JAVA®, ENPharos TP®, ENPharos LINK® and ENPharos TRACE® are registered trademark of Dabomsoft Co., Ltd. Other products, titles or services may be registered trademarks of their respective companies.

ENPharos JAVA®, ENPharos TP®, ENPharos LINK® 와 ENPharos TRACE® 는 Dabomsoft Co., Ltd.의 등록 상표입니다. 기타 모든 제품들과 회사 이름은 각각 해당 소유주의 상표로서 참조용으로만 사용됩니다.

## 안내서 정보

안내서 제목: ENPharos TP Installation Guide

발행일: 2015-08-01

소프트웨어 버전: ENPharos TP v5.0

안내서 버전: v5.0

# 목 차

1. 설치 개요.....	2
1.1. ENPharos TP 개요.....	2
1.2. 에이전트 설치 구성요소.....	2
1.2.1. 패키지 구성.....	2
1.3. 에이전트 설치 전 준비사항.....	3
1.3.1. ENPharos TP 계정 생성.....	3
1.3.2. 설치 환경 확인.....	3
1.3.3. 수집서버 설치에 대한 준비사항.....	3
2. 수집서버 설치와 제거.....	6
2.1. UNIX에서 수집서버 설치.....	6
2.1.1. 수집서버 바이너리 다운로드.....	6
2.1.1.1. 수집서버 디렉터리 생성 및 설치파일 이동.....	6
2.1.1.2. 설치파일 압축풀기.....	6
2.1.1.3. 수집서버 설치 후 디렉터리.....	6
2.1.2. 수집서버 환경 설정.....	6
2.1.3. 수집서버 기본 설정.....	8
2.1.3.1. <name>.....	8
2.1.3.2. <ip>.....	8
2.1.3.3. <targets>.....	9
2.1.3.4. <database>.....	9
2.1.3.5. <vendor>.....	10
2.1.3.6. <driver-class>.....	10
2.1.3.7. <url>.....	10
2.1.3.8. <user>.....	10
2.1.3.9. <password>.....	10
2.1.3.10. <keeping-period>.....	11
2.1.4. 카산드라 기동 및 종료.....	11
2.1.5. 수집서버 기동 및 종료.....	11
2.1.5.1. 수집서버 기동.....	11
2.1.5.2. 수집서버 종료.....	12
2.2. Windows에서 수집서버 설치.....	12
2.2.1. 수집서버 바이너리 다운로드.....	12
2.2.1.1. 수집서버 디렉터리 생성 및 설치파일 이동.....	13
2.2.1.2. 설치파일 압축풀기.....	13
2.2.1.3. 설치파일 내용 확인.....	13
2.2.2. 수집서버 환경 설정.....	13
2.2.3. 수집서버 기본 설정.....	15

2.2.3.1. <name> .....	15
2.2.3.2. <ip> .....	15
2.2.3.3. <targets> .....	16
2.1.3.4. <database> .....	16
2.1.3.5. <vendor> .....	17
2.1.3.6. <driver-class> .....	17
2.1.3.7. <url> .....	17
2.1.3.8. <user> .....	17
2.1.3.9. <password> .....	17
2.2.3.1. <keeping-period> .....	17
2.2.4. 카산드라 기동 및 종료 .....	18
2.2.5. 수집서버 기동 및 종료 .....	18
2.2.5.1. 수집서버 기동 .....	18
2.2.5.2. 수집서버 종료 .....	19
2.3. 수집서버 제거 .....	19
3. 에이전트 설치와 제거 .....	22
3.1. 에이전트 설치 .....	22
3.1.1. 에이전트 인스톨러 실행 .....	22
3.1.2. 설치 확인 .....	24
3.2. Tuxedo 환경에서 에이전트 적용 .....	25
3.2.1. ENPharos TP 관리자 계정의 환경변수 확인 .....	25
3.2.2. ENPharos TP 관리자 계정의 환경설정 추가 .....	27
3.2.2.1. ORACLE 환경 설정 .....	27
3.2.2.2. Tuxedo 환경 설정 .....	27
3.2.3. ENPharos TP 라이브러리 권한 변경 .....	28
3.2.4. Tuxedo 관리자 계정의 환경변수 등록 .....	28
3.2.5. ENPharos TP 설정 파일 수정 (pharostp.cfg) .....	30
3.2.5.1. 에이전트 정보 등록 .....	30
3.2.5.2. TP Monitor 정보 등록 .....	32
3.2.5.3. 수집서버 정보 등록 .....	33
3.2.5.4. 모니터링 정보 등록 .....	34
3.2.5.5. TUXEDO 어드민 모니터링 정보 .....	40
3.2.5.6. 모듈 모니터링 정보 .....	41
3.2.5.7. SQL 모니터링 정보 .....	41
3.2.5.8. 포지션 정보 .....	43
3.2.5.9. 에러위치 정보 .....	45
3.2.5.10. 로그 정보 .....	46
3.2.5.11. 공유메모리 정보 등록 .....	47
3.2.5.12. 연계서비스 정보 등록 .....	48
3.2.5.13. 거래코드 정보 등록 .....	48
3.2.5.14. 데몬 프로세스 정보 등록 .....	49
3.2.6. psyrmon 생성 .....	51

3.3. Tmax 환경에서 에이전트 적용 .....	51
3.3.1. ENPharos TP 관리자 계정의 환경변수 확인 .....	51
3.3.2. ENPharos TP 라이브러리 권한 변경 .....	53
3.3.3. ENPharos TP 관리자 계정의 환경설정 추가 .....	54
3.3.3.1. ORACLE 환경 설정 .....	54
3.3.3.2. Tmax 환경 설정 .....	54
3.3.4. Tmax 관리자 계정의 환경변수 등록 .....	54
3.3.5. Tmax 시스템에 tmaxadminfosvc 서버 등록 .....	56
3.3.6. ENPharos TP 관리자 계정의 tmax.env 설정 .....	56
3.3.7. ENPharos TP 설정파일 수정(pharostp.cfg) .....	57
3.3.8. psyrmon 생성 .....	57
3.4. 에이전트 기동과 종료 .....	57
3.4.1. 에이전트 기동 .....	57
3.4.2. 에이전트 종료 .....	58
3.5. UI 접속 .....	58
3.6. 에이전트 제거 .....	59

# 그림 목차

[그림] 1-1 ENPharos TP 에이전트 구성도 .....	2
[그림] 2-1 수집서버 - pharos_env.sh 의 설정.....	7
[그림] 2-2 수집서버 - pharos.xml.....	8
[그림] 2-3 <targets> 기본 설정 대상.....	9
[그림] 2-4 수집서버 - pharos.xml 의 데이터베이스 설정.....	10
[그림] 2-5 수집서버 최초 기동 화면 .....	12
[그림] 2-6 수집서버 디렉터리 구조.....	13
[그림] 2-7 수집서버 - pharos_env.cmd 의 설정 샘플.....	14
[그림] 2-8 수집서버 - pharos.xml.....	15
[그림] 2-9 <targets> 기본 설정 대상.....	16
[그림] 2-10 수집서버 - pharos.xml 의 데이터베이스 설정 .....	16
[그림] 2-11 수집서버 기동로그.....	19
[그림] 3-1 ENPharos TP 에이전트 설치 정보 .....	23
[그림] 3-2 Pharso TP 관리자 환경변수 .....	25
[그림] 3-3 pharosadt.env 설정 내용.....	25
[그림] 3-4 ORACLE 환경변수 .....	27
[그림] 3-5 Tuxedo 환경변수.....	27
[그림] 3-6 Tuxedo 계정 환경변수 .....	28
[그림] 3-7 pharostp.env 설정 내용 .....	29
[그림] 3-8 pharostp.cfg – 에이전트 정보.....	31
[그림] 3-9 pharostp.cfg – TP Monitor 정보 .....	32
[그림] 3-10 pharostp.cfg – 수집서버 정보 .....	33
[그림] 3-11 pharostp.cfg – 모니터링 정보 .....	36
[그림] 3-12 pharostp.cfg – TUXEDO 어드민 모니터링 정보.....	40
[그림] 3-13 pharostp.cfg – 모듈 모니터링 정보.....	41
[그림] 3-14 pharostp.cfg – SQL 모니터링 정보 .....	42
[그림] 3-15 pharostp.cfg – 포지션 정보 .....	43
[그림] 3-16 pharostp.cfg – 에러위치 정보 .....	45
[그림] 3-17 pharostp.cfg – 로그 정보 .....	46
[그림] 3-18 pharostp.cfg – 공유메모리 정보.....	47
[그림] 3-19 pharostp.cfg – 연계서비스 정보.....	48
[그림] 3-20 txcode.cfg – 거래코드 파일 정보 .....	48
[그림] 3-21 데몬 프로세스 파일 정보.....	49
[그림] 3-22 ENPharos TP 관리자 환경변수 .....	51
[그림] 3-23 pharosadt.env 정보.....	52
[그림] 3-24 ORACLE 환경 변수.....	54

[그림] 3-25 Tmax 환경 변수 .....	54
[그림] 3-26 Tmax 계정 환경변수 .....	55
[그림] 3-27 pharostp.env 설정 내용 .....	55
[그림] 3-28 Tmax 시스템의 서비스 등록 방법 .....	56
[그림] 3-29 ENPharos TP tmax.env 설정 .....	57
[그림] 3-30 ENPharos TP UI 로그인 화면 .....	59
[그림] 3-31 Tmax conifg – 서비스 제거 .....	60
[그림] 3-32 pharostp.env 제거 .....	60
[그림] 3-33 pharosadt.env .....	60

# 표 목차

[표] 2-1 수집서버 - pharos_env.sh 설정 항목.....	8
[표] 2-2 수집서버 - pharos_env.cmd 수정방법 및 주의사항 .....	15
[표] 3-1 ENPharos TP 에이전트 디렉터리 설명.....	24
[표] 3-2 pharosadt.env 환경 변수 설명.....	26
[표] 3-3 pharostp.env 환경 변수 설명 .....	30
[표] 3-4 pharosadt.env 파일 내용 .....	53



# 안내서에 대하여

## 안내서의 대상

본 안내서는 ENPharos TP®(이하 ENPharos TP)를 사용하여 어플리케이션 성능을 모니터링 하고자 하는 관리자를 위한 기본 설치 안내서이다. ENPharos TP 에이전트를 설치하기 위한 시스템의 기본 구성요소와 설치 전 준비사항에 대해서 설명한다. 기본 설정 중심의 설치 방법을 통해 비교적 쉽게 설치를 완료할 수 있다.

또한 ENPharos TP를 사용해서 어플리케이션 성능 모니터링의 다양한 구축 환경에 대한 설명과 해당 환경에서 적용을 위한 방법을 제시한다.

## 안내서의 전제 조건

본 안내서는 관리자가 ENPharos TP 에이전트를 개발이나 운영중인 시스템에 설치하기 위한 설치 안내서이다. 본 안내서를 원활하게 이해하기 위해서는 다음과 같은 사항을 미리 알고 있어야 한다.

- 운영체제 별 기본 명령어에 대한 기본적인 이해
- Database에 대한 기본적인 이해
- TP Monitor에 대한 기본적인 이해
- TP Service 프로그래밍의 이해

## 관련 안내서

본 안내서는 ENPharos TP 에이전트의 기본적인 설치 방법을 설명하며, ENPharos TP 운영 및 사용자 안내서와 수집서버 운영 안내서는 다루지 않는다.

- "ENPharos TP Administration Guide"

ENPharos TP의 에이전트에 대한 환경설정 파일과 에이전트 관리방법에 대해 기술한다.

- "ENPharos TP User Guide"

ENPharos TP의 각 기능설명 및 분석방법에 기술한다.

- "ENPharos Server Administration Guide"

수집서버에 대한 환경설정 파일과 시스템 관리방법에 대해 기술한다.

# 안내서 구성

ENPharos TP Installation Guide는 총 3개의 장으로 구성되어 있다. 각 장의 주요 내용은 다음과 같다.

- 제1장: 설치 개요

ENPharos TP 에이전트를 설치하기 전에 사용자가 사전에 점검하고 준비해야 할 사항들에 대해 기술한다.

- 제2장: 수집서버 설치와 제거

수집서버를 설치하고 제거하는 방법 및 기동과 종료하는 방법에 대해 기술한다.

- 제3장: 에이전트 설치와 제거

ENPharos TP 에이전트를 설치하고 제거하는 방법 및 기동과 종료하는 방법에 대해 기술한다.

# 안내서 규약

표 기	의미
<AaBbCc123>	프로그램 소스 코드의 파일명
<Ctrl>+C	Ctrl과 C를 동시에 누름
[Button]	GUI의 버튼 또는 메뉴 이름
진하게	강조
" "(따옴표)	다른 관련 안내서 또는 안내서 내의 다른 장 및 절 언급
'입력항목'	화면 UI에서 입력 항목에 대한 설명
하이퍼링크	메일계정, 웹 사이트
>	메뉴의 진행 순서
+----	하위 디렉터리 또는 파일 있음
----	하위 디렉터리 또는 파일 없음
참고	참고 또는 주의사항
[그림] 1-1	그림 이름
[표] 1-1	표 이름
AaBbCc123	명령어, 명령어 수행 후 화면에 출력된 결과물, 예제코드
[ ]	옵션 인수 값
	선택 인수 값
「」	디렉터리 경로 또는 해당 파일 경로

## 시스템 환경

항목	요구사항	
운영체제	IBM AIX 5.x, IBM AIX 6.x, IBM AIX 7.x	
	HP-UX 11.xx	
	Linux XX	
	Solaris 7~9 (SunOS 5.7~5.9)	
Hardware	RAM	2048MB 이상 메모리 공간
	Disk Space	최소 10GB
	OS	32Bit, 64Bit

## 지원 TP Monitor

제품명	요구사항
Tmax	Tmax 3.0, Tmax 4.0, Tmax 5.0
Tuxedo	Tuxedo 8~11gR1

# 01

---

## 설치 개요

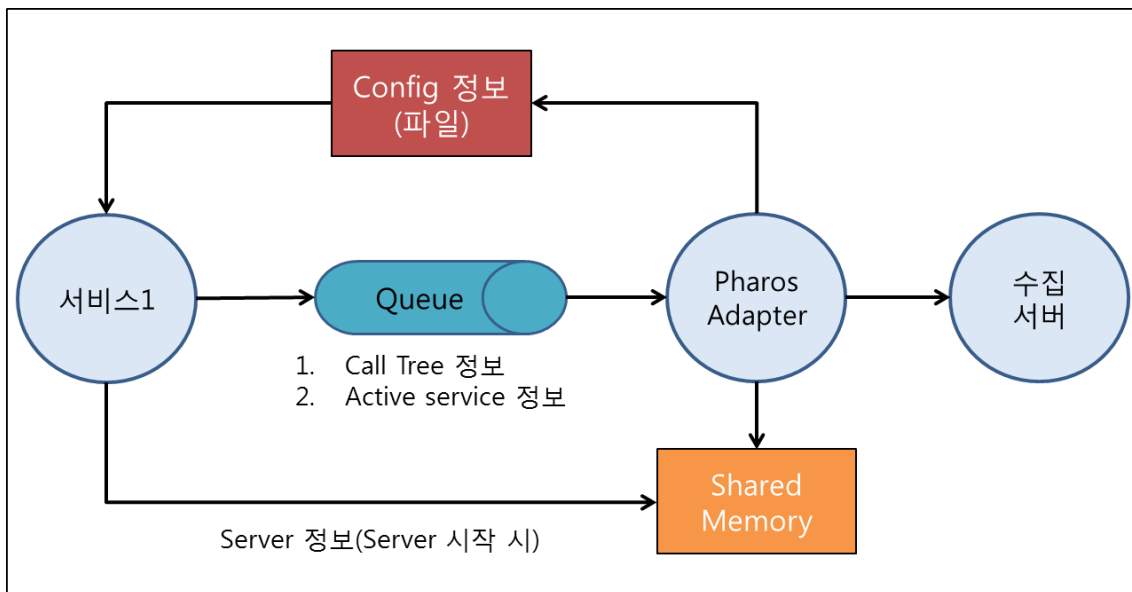
- 1.1 ENPharos TP 개요
  - 1.2 에이전트 설치 구성요소
  - 1.3 에이전트 설치 전 준비사항
-

# 1. 설치 개요

본 장에서는 ENPharos TP 개요, 구성요소 및 설치 전 준비사항에 대해 기술한다.

## 1.1. ENPharos TP 개요

ENPharos TP는 Middleware 시스템 하에서 실행되는 서비스가 호출되는 시점부터 종료될 때까지 실시간으로 서비스 처리 상태에 대한 Call-Tree 및 데이터베이스 처리 정보를 수집하여 서비스가 현재 어떠한 상태에 놓여 있는지를 파악할 수 있도록 제공하는 시스템이다. 또한 서비스 처리 시 CPU 사용량, 메모리 Leak, 경과시간을 추적하여 서비스가 안정적으로 실행되는지의 정보를 제공하는 시스템이다.



[그림] 1-1 ENPharos TP 에이전트 구성도

## 1.2. 에이전트 설치 구성요소

ENPharos TP 에이전트 설치 구성 요소는 다음과 같다.

### 1.2.1. 패키지 구성

ENPharos TP 에이전트의 Package 구성은 다음과 같다.

#### ● ENPharos TP 설치 구성

1. ENPharos TP Agent
2. Pharos Server

## ● ENPharos TP 안내서

1. ENPharos TP Installation Guide
2. ENPharos TP Administration Guide
3. ENPharos TP User Guide
4. Pharos Server Administration Guide

## 1.3. 에이전트 설치 전 준비사항

ENPharos TP 에이전트는 별도의 Installer를 사용한 설치 방식이 아닌 바이너리 형태의 파일을 Unix 또는 Linux 운영체제에서 실행하고 설정을 조정함으로써 간단하게 설치할 수 있다. ENPharos TP 에이전트는 Unix(Linux)를 지원하고 수집서버는 Unix(Linux)와 Windows를 지원한다. 환경변수 설정이나 기동파일 실행 시 운영체제에 맞는 파일 형식을 사용하여야 한다.

본 ENPharos TP Installation Guide에서는 ENPharos TP와 수집서버의 설치와 관련된 기본 설정만을 다루고 있다. ENPharos TP 에 관한 세부적인 설정 방법이나 최적화 방법은 "ENPharos TP Administration Guide"를 참고하고 수집서버는 "Pharos Server Administration Guide"를 참조한다.

### 1.3.1. ENPharos TP 계정 생성

TP Monitor 관리 계정으로 설치해도 되지만, 보통 ENPharos TP 관리자 계정을 따로 만들 것을 권장한다.

### 1.3.2. 설치 환경 확인

ENPharos TP 에이전트의 최소 요구 사항에 합당한지 확인하고, 설치 스크립트를 실행하기 전에 시스템에 대해 몇 가지 확인해야 할 사항이 있다. 서버에 설치된 운영체제와 아키텍처(32bit or 64bit), ENPharos TP 를 설치하고자 하는 디렉터리의 절대 경로, 그리고 설치 후 테스트를 위해 서버의 IP 주소를 미리 알고 있어야 한다.

### 1.3.3. 수집서버 설치에 대한 준비사항

"1. 설치개요" 앞의 3 페이지에 있는 시스템 환경을 참조한다.





# 02

---

## 수집서버 설치와 제거

- 2.1 UNIX에서 수집서버 설치
  - 2.2 Windows에서 수집서버 설치
  - 2.3 수집서버 제거
-

## 2. 수집서버 설치와 제거

본 장에서는 수집서버를 설치하고 제거 하는 방법에 대해 기술한다.

### 2.1. UNIX에서 수집서버 설치

이해를 돕기 위해 UNIX 운영체제 환경에서 설명한다. Windows계열 운영체제 환경에서 설치 방법은 "2.2. Windows 에서 수집서버 설치"를 참조한다.

#### 2.1.1. 수집서버 바이너리 다운로드

Pharos 수집서버 설치파일은 온라인 <http://www.dabomsoft.com> 및 오프라인에서 제공한다. Windows 운영체제에서 설치 되어야 할 곳에 수집서버 디렉터리를 생성한 후 제공된 설치파일을 복사하고 압축을 해제한다.

##### 2.1.1.1. 수집서버 디렉터리 생성 및 설치파일 이동

```
[WAS@hostname:/home]mkdir PharosServer
[WAS@hostname:/home]move pharos-server-bundle-4.0.X.jar PharosServer
```

기본적으로 수집서버 폴더의 이름은 변경이 가능하다. 또한 제공된 설치파일의 파일명도 버전에 따라 파일명이 변경될 수 있다. 본 설치 안내서에서는 이해를 쉽게 하기 위해 임의로 기본 설치 폴더를 「/home/PharosServer」로 정한다.

##### 2.1.1.2. 설치파일 압축풀기

```
[WAS@hostname:/home]cd PharosServer
[WAS@hostname:/home/PharosServer]jar xvf pharos-server-bundle-4.0.X.jar
```

이와 같이 압축을 해제하면 수집서버 설치가 완료된다.

##### 2.1.1.3. 수집서버 설치 후 디렉터리

정상적으로 설치된 수집서버 홈 디렉터리의 하위 디렉터리들과 실행파일들이다.

```
[WAS@hostname:/home]cd PharosServer
[WAS@hostname:/home/PharosServer]ls
./                sleep.exe                stop_pharos.sh*
../              ReleaseNotes/            start_pharos.cmd
BASE_HOME/       cassandra/               UI/                   start_pharos.sh*
META-INF/        pharos-server-bundle-4.0.0.jar  stop_pharos.cmd
```

### 2.1.2. 수집서버 환경 설정

수집서버의 환경을 설정하기 위해 pharos\_env.sh 파일을 열어 수정한다.

```
[WAS@hostname:/home/PharosServer/UI/bin]vi pharos_env.sh
```

파일을 열면 [그림] 2-1와 같은 내용을 확인 할 수 있다.

```

JAVA_HOME=/usr/java17; export JAVA_HOME

PHAROS_ROOT=/home/oldman/PharosServer/PharosServer35_DEV; export PHAROS_ROOT

PHAROS_BASE=$PHAROS_ROOT/BASE_HOME; export PHAROS_BASE
PHAROS_HOME=$PHAROS_ROOT/UI; export PHAROS_HOME

WEB_APP_BASE=$PHAROS_BASE/webapps; export WEB_APP_BASE
#### Available WEB_DD_FILE ####
#all-in-one : web.xml
#ui-only    : web-uionly.xml
#java-only  : web-java.xml
#tp-only    : web-tp.xml
#link-only  : web-link.xml
#webserver-only : web-webserver.xml
#daemon-only : web-daemon.xml
WEB_DD_FILE=$PHAROS_BASE/webapps/ROOT/WEB-INF/web.xml; export WEB_DD_FILE

#### PORT CONFIG ####
PORT_TOMCAT_HTTP=45000; export PORT_TOMCAT_HTTP
PORT_TOMCAT_SHUTDOWN=45005; export PORT_TOMCAT_SHUTDOWN

PORT_UI_PUSH=45001; export PORT_UI_PUSH
PORT_AGENT=44001; export PORT_AGENT
PORT_FLASH_POLICY=10843; export PORT_FLASH_POLICY
#####
----- <이하 생략> -----

```

[그림] 2-1 수집서버 - pharos\_env.sh 의 설정

pharos\_env.sh 파일은 수집서버에서 사용하는 환경변수를 설정하는 파일이다. 환경변수는 설치한 시스템에 맞게 조정할 필요가 있으며, 수정할 환경변수의 변경방법 및 주의사항은 [표] 2-1과 같다.

환경변수	설명
JAVA_HOME	수집서버가 사용할 JDK 경로를 지정한다. JAVA_HOME=/usr/java17
PHAROS_ROOT	수집서버의 ROOT 경로를 지정한다. PHAROS_ROOT=/home/oldman/PharosServer/PharosServer35_DEV
PHAROS_BASE	수집서버의 공통 모듈 경로를 지정한다. PHAROS_BASE=\$PHAROS_ROOT/BASE_HOME
PHAROS_HOME	수집서버 엔진 위치를 지정한다. PHAROS_HOME=\$PHAROS_ROOT/UI
WEB_APP_BASE	수집서버 웹 애플리케이션(라이브러리, UI 등)의 경로를 지정한다. WEB_APP_BASE=\$PHAROS_BASE/webapps
WEB_DD_FILE	수집서버 배포서술자의 경로를 지정한다.(web.xml 파일 경로) 서버 분산하는 경우 각 서버에 맞게 WEB_DD_FILE을 지정한다. WEB_DD_FILE=\$PHAROS_BASE/webapps/ROOT/WEB-INF/web.xml
PORT_TOMCAT_HTTP	톰캣 HTTP 접속 포트를 지정한다. (방화벽 오픈 필요함.) PORT_TOMCAT_HTTP=45000
PORT_TOMCAT_SHUTDOWN	톰캣 Shutdown 포트를 지정한다. PORT_TOMCAT_SHUTDOWN=45005

PORT_UI_PUSH	신규 UI에서 PUSH 시에 사용하는 포트를 지정한다. PORT_UI_PUSH=45001
PORT_AGENT	에이전트와 접속할 포트번호를 지정한다. PORT_AGENT=44001
PORT_FLASH_POLICY	플래쉬의 보안을 위해 policy파일을 내려 받을 시 사용된다. PORT_FLASH_POLICY=10843

[표] 2-1 수집서버 - pharos\_env.sh 설정 항목

## 참고

Windows 운영체제와 UNIX(Linux)의 디렉터리 경로지정 방식의 차이를 확인 후 수정해야 한다.  
디렉터리 경로에 공백이 있는 경우 "" 을 이용하여 경로를 설정한다.

### 2.1.3. 수집서버 기본 설정

수집서버의 설정을 시스템에 맞게 변경하기 위해서는 아래 [그림] 2-2 와 같이 「/home/PharosServer/UI/config/pharos.xml」의 기본 설정 값을 변경한다.

```
<?xml version="1.0" encoding="UTF-8"?>
<server>
  <name>PharosServer</name>
  <ip>127.0.0.1</ip>
  <targets>
    <target>UiServer</target>
    <!--target>J2EE:10:44001</target-->
    <target>TPMonitor:10:44001</target>
    <!--target>AnyLink:10:44001</target-->
    <target>Webserver:10:44001</target>
    <target>Daemon:10:44001</target-->
  </targets>
  <database>
    <statistics>
      <vendor>PostgreSQL</vendor>
      <driver-class>org.postgresql.Driver</driver-class>
      <url>jdbc:postgresql://localhost:5432/pharos</url>
      <user>pharos</user>
      <password>2g772AXN</password>
    </statistics>
    <raw-data>
      <vendor>Cassandra</vendor>
      <keeping-period>7</keeping-period>
    </raw-data>
  </database>
</server>
```

[그림] 2-2 수집서버 - pharos.xml

#### 2.1.3.1. <name>

<name> 태그는 수집서버의 이름을 설정한다.

```
<name>PharosServer</name>
```

#### 2.1.3.2. <ip>

<ip> 태그는 수집서버가 설치된 시스템의 IP Address를 설정 한다. 설정 값으로 IPv4 형식으로 입력하며, 설정된 IP Address를 이용하여 에이전트, UI가 데이터 통신을 한다.

```
<ip>127.0.0.1</ip>
```

### 2.1.3.3. <targets>

<targets> 태그는 수집될 데이터 엔진모듈 종류와 에이전트 접속 포트를 지정한다. 하위 태그인 <target> 태그에서 수집될 데이터 별로 엔진모듈이 추가 되며 커넥션 개수, 커넥션 포트를 표기하여야 한다.

```
<targets>
  <target>UiServer</target>
  <!--target>J2EE:10:44001</target-->
  <target>TPMonitor:10:44001</target>
  <!--target>AnyLink:10:44001</target>
  <target>Webserver:10:44001</target>
  <target>Daemon:10:44001</target-->
</targets>
```

[그림] 2-3 <targets> 기본 설정 대상

<target>J2EE:10:44001</target>에서 J2EE는 대상 엔진모듈, 10은 UiServer와의 커넥션개수, 44001은 UiServer와의 커넥션 포트를 의미한다.

수집서버 설치 시 <target> 태그는 기본설정이 J2EE로 되어 있다. TP Monitor와 관련된 수집서버를 설치 하기 위해서는 <target> 태그를 TPMonitor로 변경해야 한다.

```
<targets>
  <target>UiServer</target>
  <target>TPMonitor:10:44001</target>
</targets>
```

### 2.1.3.4. <database>

<database>태그는 수집서버 데이터베이스를 설정하는 태그이다. 수집서버의 데이터베이스는 두 가지 종류로 구성된다.

- <statistics> : 통계 데이터 저장 설정
- <raw-data> : 실시간 데이터 및 개별 요청 정보 저장 설정

설치 시 기본 설정은 내장되어 있는 Cassandra DB를 대상으로 설치와 관련된 <database>태그 설정법만 확인하며 설명하며 자세한 사항은 "Pharos Server Administration Guide"를 참고한다.

pharos.xml 파일 내용 중 [그림] 2-4와 같이 <server>태그 안에 <database>태그를 확인한다.

```

----- <생략> -----
    <database>
      <statistics>
        <vendor>PostgreSQL</vendor>
        <driver-class>org.postgresql.Driver</driver-class>
        <url>jdbc:postgresql://localhost:5432/pharos</url>
        <user>pharos</user>
        <password>2g772AXN</password>
      </statistics>
      <raw-data>
        <vendor>Cassandra</vendor>
        <keeping-period>7</keeping-period>
      </raw-data>
    </database>
----- <생략> -----

```

[그림] 2-4 수집서버 - pharos.xml 의 데이터베이스 설정

### 2.1.3.5. <vendor>

<vendor>의 사용할 데이터베이스의 명칭을 설정한다. 기본값으로 PostgreSQL이 설정되어 있다.

```
<vendor>PostgreSQL</vendor>
```

### 2.1.3.6. <driver-class>

<driver-class>태그는 <vendor>태그에 설정된 DB의 드라이버 클래스를 설정한다. 기본 값으로 PostgreSQL의 드라이버 클래스가 설정되어 있다.

```
<driver-class>org.postgresql.Driver</driver-class>
```

### 2.1.3.7. <url>

<url>은 statistics-database의 데이터 저장 경로를 설정하는 곳으로, PostgreSQL 데이터베이스를 사용 하면, statistics-database는 「jdbc:postgresql://{데이터베이스 원격 접속 IP}:{사용포트}/{SID}」로 설정한다.

```
jdbc:postgresql://localhost:5432/pharos
```

### 2.1.3.8. <user>

<user>는 PostgreSQL 데이터베이스에 접속할 사용자의 ID를 입력한다.

```
<user>pharos</user>
```

### 2.1.3.9. <password>

<password> 태그는 데이터베이스에 접속할 사용자 ID의 비밀번호를 입력한다. <password>의 값은 암호화 과정을 거쳐서 설정 한다. <password> 암호화는 아래와 같이 변경 한다.

```
[WAS@hostname:/home/PharosServer]cd /home/PharosServer/UI/bin
[WAS@hostname:/home/PharosServer/UI]pharos_pw.sh pharos
Encoded password : [2g772AXN]
```

pharos\_pw.sh 명령어를 이용하여 매개변수로 암호를 지정하면 암호화 할 수 있다. pharos.xml에서는 필수 적으로 암호화된 값을 사용해야 한다.

```
<password>2g772AXN</password>
```

### 2.1.3.10. <keeping-period>

<keeping-period> 는 raw-database로 사용하는 카산드라에 저장된 데이터들의 최대 보관 일자를 지정하는 부분이다. 기본값으로 7이 설정되어 있으며, "생성 후 7일이 지난 데이터는 삭제한다." 라는 의미가 된다. <keeping-period>는 시스템 디스크 용량에 영향을 줄 수 있으므로, 디스크 사이즈를 감안하여 적정수치를 입력해야 한다. 기본적인 설치 시에는 권장 값인 7을 그대로 사용하도록 한다.

```
<keeping-period>7</keeping-period>
```

---

#### 참고

수집서버가 통계데이터를 저장하는 데이터베이스는 제품 내 포함되어 있는 메모리DB인 H2 데이터베이스와 상용 RDBMS인 Oracle(8i, 9i, 10g, 11g) 데이터베이스, PostgreSQL 데이터베이스를 지원 한다.

Oracle 데이터베이스, H2 데이터베이스 연결정보는 config 디렉터리 내 sample\_pharos.xml에서 예제가 제공된다.

---

### 2.1.4. 카산드라 기동 및 종료

<raw-data> 의 생성을 위해 카산드라를 실행해야 한다. 카산드라를 실행하기 위해서는 수집서버 홈 디렉터리 「/cassandra/bin」에 위치하는 cassandra 파일을 실행한다.

```
[WAS@hostname:/home/PharosServer]cd cassandra/bin
[WAS@hostname:/home/PharosServer/cassandra/bin]cassandra
```

### 2.1.5. 수집서버 기동 및 종료

수집서버의 기동, 종료 명령어는 수집서버가 홈 디렉터리 「/home/PharosServer/UI/bin」에 위치하는 start.sh, stop.sh 이다. 수집서버를 분리할 경우에는 「/home/PharosServer」에 위치하는 start\_pharos.sh, stop\_pharos.sh 파일로 기동 및 종료한다. 상세 설명은 "Pharos Server Administration Guide" 의 "A. 수집서버 분리"를 참조한다.

#### 2.1.5.1. 수집서버 기동

「start.sh」 파일을 실행하면 수집서버가 기동 된다.

```
[WAS@hostname:/home]cd PharosServer/UI/bin
[WAS@hostname:/home/PharosServer/UI/bin]start.sh
```

```

2012. 7. 31 오전 11:38:36 org.apache.tomcat.util.net.NioSelectorPool getSharedSelector
정보: Using a shared selector for servlet write/read
2012. 7. 31 오전 11:38:36 org.apache.coyote.http11.Http11NioProtocol init
정보: Initializing Coyote HTTP/1.1 on http-45000
2012. 7. 31 오전 11:38:36 org.apache.catalina.startup.Catalina load
정보: Initialization processed in 243 ms
2012. 7. 31 오전 11:38:36 org.apache.catalina.core.StandardService start
정보: Starting service Catalina
2012. 7. 31 오전 11:38:36 org.apache.catalina.core.StandardEngine start
정보: Starting Servlet Engine: Apache Tomcat/6.0.26
2012. 7. 31 오전 11:38:36 org.apache.catalina.startup.HostConfig deployDescriptor
정보: Deploying configuration descriptor ROOT.xml
TARGET: UISEWER
TARGET: J2EE:20:44001

*****
Pharos Server Information.
-----
VERSION           : 3.5 <3.5.7-SNAPSHOT>
OS                : Windows 7 6.1 amd64
JAVA_HOME         : C:/Java/jdk1.6.0_33
PHAROS_BASE       : C:/PharosServer/BASE_HOME
PHAROS_HOME       : C:/PharosServer/UI
SERVER TYPE       : master
SERVER NAME       : myServer
SERVER IP         : 127.0.0.1
PORT [agent]      : 44001
PORT [ui]         : 44000
DB [main]         : H2
DB [rawdata]      : H2
-----
compressDayPassed : 7
rawdataKeepingPeriod : 70
summarySaveLimit   : 0
blobSaveLimit      : 0
blobForceSaveIssue : true
blobUseCompress    : true
-----
useUriWithParams   : true
useBizGroups       : true
*****

Processing to create schemas....
Schemas for Main DB[javal] was generated(version = 20120622).
Schemas for Main DB[daemon] was generated(version = 20111030).
Schemas for Main DB[link] was generated(version = 20111030).
Schemas for Main DB[tpl] was generated(version = 20111030).
Schemas for Main DB[txl] was generated(version = 20120607).
Processing to insert initial data ....
Initial datas successfully inserted.
2012. 7. 31 오전 11:38:53 com.kwise.pharos.server.report.ReportConfig initLogger
정보: Logger(com.kwise.pharos.server.report) is initialized.
2012. 7. 31 오전 11:38:58 org.apache.coyote.http11.Http11NioProtocol start
정보: Starting Coyote HTTP/1.1 on http-45000
2012. 7. 31 오전 11:38:58 org.apache.catalina.startup.Catalina start
정보: Server startup in 22004 ms

```

[그림] 2-5 수집서버 최초 기동 화면

### 2.1.5.2. 수집서버 종료

「stop.sh」 파일을 실행하면 수집서버가 종료 된다.

```

[WAS@hostname:/home]cd PharosServer/UI/bin
[WAS@hostname:/home/PharosServer/UI/bin]stop.sh

```

## 2.2. Windows에서 수집서버 설치

이해를 돕기 위해 Windows 운영체제 환경에서 설명한다. UNIX 계열 운영체제 환경에서 설치 방법은 “2.1. UNIX 에서 수집서버 설치” 를 참조한다.

### 2.2.1. 수집서버 바이너리 다운로드

Pharos 수집서버 설치파일은 온라인 <http://technet.Dabomsoft.com> 및 오프라인에서 제공한다. Windows 운영체제에서 설치 되어야 할 곳에 수집서버 디렉터리를 생성한 후 제공된 설치파일을 복사하고 압축을 해제한다.



### 2.2.1.1. 수집서버 디렉터리 생성 및 설치파일 이동

```
C:\>mkdir PharosServer  
C:\>move pharos-server-bundle-4.0.X.jar PharosServer
```

기본적으로 수집서버 폴더의 이름은 변경이 가능하다. 또한 제공된 설치파일의 파일명도 버전에 따라 파일명이 변경될 수 있다.

본 설치 안내서에서는 이해를 쉽게 하기 위해 임의로 기본 설치 폴더를 「C:\PharosServer」로 정한다.

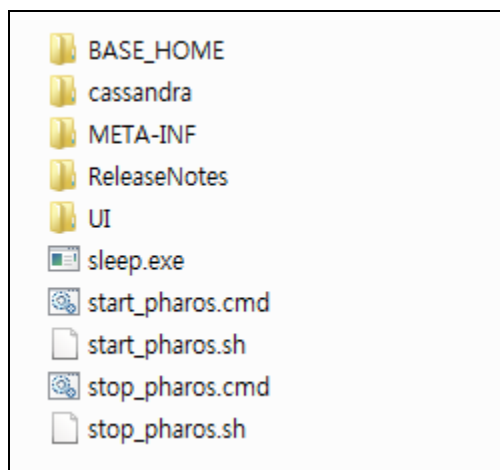
### 2.2.1.2. 설치파일 압축풀기

```
C:\>cd PharosServer  
C:\PharosServer>jar xvf pharos-server-bundle-4.0.X.jar
```

이와 같이 압축을 해제하면 수집서버 설치가 완료된다.

### 2.2.1.3. 설치파일 내용 확인

아래 [그림] 2-6의 항목들은 정상적으로 설치된 수집서버 홈 디렉터리의 하위 디렉터리들과 실행파일들이다.



[그림] 2-6 수집서버 디렉터리 구조

## 2.2.2. 수집서버 환경 설정

설치가 완료되면 수집서버의 환경을 설정하기 위해 「pharos\_env.cmd」 파일을 수정한다.

```
C:\PharosServer\UI\bin>pharos_env.cmd
```

파일을 열면 [그림] 2-7과 같은 내용을 확인 할 수 있다.

```

SET JAVA_HOME=${win.java.home}

SET PHAROS_ROOT=${win.pharos.root}

SET PHAROS_BASE=%PHAROS_ROOT%/BASE_HOME
SET PHAROS_HOME=%PHAROS_ROOT%/UI

SET WEB_APP_BASE=%PHAROS_BASE%/webapps
REM ##### Available WEB_DD_FILE #####
REM all-in-one : web.xml
REM ui-only    : web-uionly.xml
REM java-only   : web-java.xml
REM tp-only    : web-tp.xml
REM link-only   : web-link.xml
REM webserver-only : web-webserver.xml
REM daemon-only : web-daemon.xml
SET WEB_DD_FILE=%PHAROS_BASE%/webapps/ROOT/WEB-INF/web.xml

REM ##### PORT CONFIG #####
SET PORT_TOMCAT_HTTP=45000
SET PORT_TOMCAT_SHUTDOWN=45005

SET PORT_UI_PUSH=45001
SET PORT_AGENT=44001
SET PORT_FLASH_POLICY=10843
REM #####
----- <이하 생략> -----

```

[그림] 2-7 수집서버 - pharos.env.cmd 의 설정 샘플

pharos.env.cmd 파일은 수집서버에서 사용하는 환경변수를 설정하는 파일이다. 환경변수는 설치한 시스템에 맞게 조정할 필요가 있으며, 수정할 환경변수의 변경방법 및 주의사항은 [표] 2-2과 같다.

환경변수	설명
JAVA_HOME	수집서버가 사용할 JDK 경로를 지정한다. JAVA_HOME=\${win.java.home}
PHAROS_ROOT	수집서버의 ROOT 경로를 지정한다. PHAROS_ROOT=\${win.pharos.root}
PHAROS_BASE	수집서버의 공통 모듈 경로를 지정한다. PHAROS_BASE=%PHAROS_ROOT%/BASE_HOME
PHAROS_HOME	수집서버 엔진 위치를 지정한다. PHAROS_HOME=%PHAROS_ROOT%/UI
WEB_APP_BASE	수집서버 웹 애플리케이션(라이브러리, UI 등)의 경로를 지정한다. WEB_APP_BASE=%PHAROS_BASE%/webapps
WEB_DD_FILE	수집서버 배포서술자의 경로를 지정한다.(web.xml 파일 경로) 서버 분산하는 경우 각 서버에 맞게 WEB_DD_FILE을 지정한다. WEB_DD_FILE=%PHAROS_BASE%/webapps/ROOT/WEB-INF/web.xml
PORT_TOMCAT_HTTP	톰캣 HTTP 접속 포트를 지정한다. (방화벽 오픈 필요함.) PORT_TOMCAT_HTTP=45000
PORT_TOMCAT_SHUTDOWN	톰캣 Shutdown 포트를 지정한다. PORT_TOMCAT_SHUTDOWN=45005
PORT_UI_PUSH	신규 UI에서 PUSH 시에 사용하는 포트를 지정한다. PORT_UI_PUSH=45001

PORT_AGENT	에이전트와 접속할 포트번호를 지정한다. PORT_AGENT=44001
PORT_FLASH_POLICY	플래쉬의 보안을 위해 policy파일을 내려 받을 시 사용된다. PORT_FLASH_POLICY=10843

[표] 2-2 수집서버 - pharos\_env.cmd 수정방법 및 주의사항

## 참고

Windows 운영체제와 UNIX(Linux)의 디렉터리 경로지정 방식의 차이를 확인 후 수정해야 한다.  
디렉터리 경로에 공백이 있는 경우 "" 을 이용하여 경로를 설정한다.

### 2.2.3. 수집서버 기본 설정

수집서버의 설정을 시스템에 맞게 변경하기 위해서는 아래 [그림] 2-8과 같이  
「C:\PharosServer\UI\config\pharos.xml」 파일의 기본 설정 값을 변경한다.

```
<?xml version="1.0" encoding="UTF-8"?>
<server>
  <name>PharosServer</name>
  <ip>127.0.0.1</ip>
  <targets>
    <target>UiServer</target>
    <!--target>J2EE:10:44001</target-->
    <target>TPMonitor:10:44001</target>
    <!--target>AnyLink:10:44001</target>
    <target>Webserver:10:44001</target>
    <target>Daemon:10:44001</target-->
  </targets>
  <database>
    <statistics>
      <vendor>H2</vendor>
      <driver-class>org.h2.Driver</driver-class>
      <url>jdbc:h2:${pharos.base}/database/main/main</url>
      <user>pharos</user>
      <password>2g772AXN</password>
    </statistics>
    <raw-data>
      <vendor>Cassandra</vendor>
      <keeping-period>7</keeping-period>
    </raw-data>
  </database>
</server>
```

[그림] 2-8 수집서버 - pharos.xml

#### 2.2.3.1. <name>

<name> 태그는 수집서버의 이름을 설정한다.

```
<name>PharosServer</name>
```

#### 2.2.3.2. <ip>

<ip> 태그는 수집서버가 설치된 시스템의 IP Address를 설정 한다. 설정 값으로 IPv4 형식으로 입력하며,  
설정된 IP Address를 이용하여 에이전트, UI가 데이터 통신을 한다.

```
<ip>127.0.0.1</ip>
```

### 2.2.3.3. <targets>

<targets> 태그는 수집될 데이터 엔진모듈 종류와 에이전트 접속 포트를 지정한다. 하위 태그인 <target> 태그에서 수집될 데이터 별로 엔진모듈이 추가 되며 커넥션 개수, 커넥션 포트를 표기하여야 한다.

```
<targets>
  <target>UiServer</target>
  <!--target>J2EE:10:44001</target-->
  <target>TPMonitor:10:44001</target>
  <!--target>AnyLink:10:44001</target-->
  <target>Webserver:10:44001</target>
  <target>Daemon:10:44001</target-->
</targets>
```

[그림] 2-9 <targets> 기본 설정 대상

<target>J2EE:10:44001</target>에서 J2EE는 대상 엔진모듈, 10은 UiServer와의 커넥션개수, 44001은 UiServer와의 커넥션 포트를 의미한다.

수집서버 설치 시 <target> 태그는 기본설정이 J2EE로 되어 있다. TP Monitor와 관련된 수집서버를 설치 하기 위해서는 <target> 태그를 TPMonitor로 변경해야 한다.

```
<targets>
  <target>UiServer</target>
  <target>TPMonitor:10:44001</target>
</targets>
```

### 2.1.3.4. <database>

<database>태그는 수집서버 데이터베이스를 설정하는 태그이다. 수집서버의 데이터베이스는 두 가지 종류로 구성된다.

- <statistics> : 통계 데이터 저장 설정
- <raw-data> : 실시간 데이터 및 개별 요청 정보 저장 설정

설치 시 기본 설정은 내장되어 있는 Cassandra DB를 대상으로 설치와 관련된 <database>태그 설정법만 확인하며 설명하며 자세한 사항은 "Pharos Server Administration Guide"를 참고한다.

pharos.xml 파일 내용 중 [그림] 2-10과 같이 <server>태그 안에 <database>태그를 확인한다.

```
----- <생략> -----
  <database>
    <statistics>
      <vendor>PostgreSQL</vendor>
      <driver-class>org.postgresql.Driver</driver-class>
      <url>jdbc:postgresql://localhost:5432/pharos</url>
      <user>pharos</user>
      <password>2g772AXN</password>
    </statistics>
    <raw-data>
      <vendor>Cassandra</vendor>
      <keeping-period>7</keeping-period>
    </raw-data>
  </database>
----- <생략> -----
```

[그림] 2-10 수집서버 - pharos.xml 의 데이터베이스 설정

### 2.1.3.5. <vendor>

<vendor>의 사용할 데이터베이스의 명칭을 설정한다. 기본값으로 PostgreSQL이 설정되어 있다.

```
<vendor>PostgreSQL</vendor>
```

### 2.1.3.6. <driver-class>

<driver-class>태그는 <vendor>태그에 설정된 DB의 드라이버 클래스를 설정한다. 기본 값으로 PostgreSQL의 드라이버 클래스가 설정되어 있다.

```
<driver-class>org.postgresql.Driver</driver-class>
```

### 2.1.3.7. <url>

<url>은 statistics-database의 데이터 저장 경로를 설정하는 곳으로, 기본 내장 데이터베이스를 사용 하면, statistics-database는 「jdbc:postgresql://{데이터베이스 원격 접속 IP}:{사용포트}/{SID}」로 설정한다.

```
jdbc:postgresql://localhost:5432/pharos
```

### 2.1.3.8. <user>

<user>는 PostgreSQL데이터베이스에 접속할 사용자의 ID를 입력한다.

```
<user>pharos</user>
```

### 2.1.3.9. <password>

<password> 태그는 데이터베이스에 접속할 사용자 ID의 비밀번호를 입력한다. <password>의 값은 암호화 과정을 거쳐서 설정 한다. <password> 암호화는 아래와 같이 변경 한다.

```
[WAS@hostname:/home/PharosServer]cd /home/PharosServer/UI/bin
[WAS@hostname:/home/PharosServer/UI]pharos_pw.cmd pharos
Encoded password : [2g772AXN]
```

pharos\_pw.sh 명령어를 이용하여 매개변수로 암호를 지정하면 암호화 할 수 있다. pharos.xml에서는 필수 적으로 암호화된 값을 사용해야 한다.

```
<password>2g772AXN</password>
```

### 2.2.3.1. <keeping-period>

<keeping-period> 는 raw-database로 사용하는 카산드라에 저장된 데이터들의 최대 보관 일자를 지정하는 부분이다. 기본값으로 7이 설정되어 있으며, "생성 후 7일이 지난 데이터는 삭제한다." 라는 의미가 된다. <keeping-period>는 시스템 디스크 용량에 영향을 줄 수 있으므로, 디스크 사이즈를 감안하여 적정수치를 입력해야 한다. 기본적인 설치 시에는 권장 값인 7을 그대로 사용하도록 한다.

```
<keeping-period>7</keeping-period>
```

---

## 참고

수집서버가 통계데이터를 저장하는 데이터베이스는 제품 내 포함되어 있는 메모리DB인 H2 데이터베이스와 상용 RDBMS인 Oracle(8i, 9i, 10g, 11g) 데이터베이스, PostgreSQL 데이터베이스를 지원 한다.

Oracle 데이터베이스, H2 데이터베이스 연결정보는 config 디렉터리 내 sample\_pharos.xml에서 예제가 제공된다.

---

### 2.2.4. 카산드라 기동 및 종료

<raw-data> 의 생성을 위해 카산드라를 실행해야 한다. 카산드라를 실행하기 위해서는 수집서버 홈 디렉터리 「%cassandra%bin」에 위치하는 cassandra.bat 파일을 실행한다.

```
C:\cd PharosServer\cassandra\bin
C:\PharosServer\cassandra\bin>cassandra.bat
```

### 2.2.5. 수집서버 기동 및 종료

수집서버 설치 및 환경 설정이 완료되면 수집서버 기동이 가능하다. 수집서버의 기동, 종료 명령어는 수집서버가 홈 디렉터리 「C:\PharosServer」에 위치하는 start\_pharos.cmd, stop\_pharos.cmd 이다.

수집서버를 분리할 경우에는 「C:\PharosServer\UI\bin」에 위치하는 start\_pharos.cmd, stop\_pharos.cmd 파일로 기동 및 종료한다. 상세 설명은 "Pharos Server Administration Guide" 의 A. 수집서버 분리를 참조한다.

#### 2.2.5.1. 수집서버 기동

「start.cmd」 실행하면 수집서버가 기동 된다.

```
C:\cd PharosServer\UI\bin
C:\PharosServer\UI\bin>start.cmd
```

```

Pharos 4.0.0
4월 30, 2013 7:55:45 오후 org.apache.tomcat.util.net.NioSelectorPool getSharedSe
lector
INFO: Using a shared selector for servlet write/read
4월 30, 2013 7:55:45 오후 org.apache.coyote.http11.Http11NioProtocol init
INFO: Initializing Coyote HTTP/1.1 on http-45000
4월 30, 2013 7:55:45 오후 org.apache.catalina.startup.Catalina load
INFO: Initialization processed in 903 ms
4월 30, 2013 7:55:45 오후 org.apache.catalina.core.StandardService start
INFO: Starting service Catalina
4월 30, 2013 7:55:45 오후 org.apache.catalina.core.StandardEngine start
INFO: Starting Servlet Engine: Apache Tomcat/6.0.26
4월 30, 2013 7:55:46 오후 org.apache.catalina.startup.HostConfig deployDescripto
r
INFO: Deploying configuration descriptor ROOT.xml
TARGET: USERVER
TARGET: J2EE:20:44001

*****
Pharos Server Information.
*****
VERSION           : 3.5 <4.0.0.130427>
OS                : Windows 8 6.2 amd64
JAVA_HOME         : C:\Program Files\Java\jdk1.7.0_17
PHAROS_BASE       : C:\PharosServer\BASE_HOME
PHAROS_HOME       : C:\PharosServer\UI
SERVER_TYPE       : master
SERVER_NAME       : PharosServer
SERVER_IP         : 
*****
PORT [AGENT]      : 44001
PORT [HTTP]      : 45000
PORT [UI-PUSH<NEW>] : 45001
PORT [Flash-Policy] : 10843
*****
DB [main]        : H2
DB [rawdata]     : Cassandra
*****
rawdataKeepingPeriod : 7
*****
useBizGroups      : true
*****
Main DB[daemon] will be updated from 20121130 to 20130430...

```

[그림] 2-11 수집서버 기동로그

## 2.2.5.2. 수집서버 종료

「stop.cmd」 실행하면 수집서버가 종료된다.

```

C:\>cd PharosServer\UI\bin
C:\PharosServer\UI\bin>stop.cmd

```

## 2.3. 수집서버 제거

수집서버는 설치 경로의 파일을 제외한 별도의 파일을 생성하지 않는다. 따라서, 수집서버가 설치된 디렉터리를 삭제하여 제거한다.





# 03

---

## 에이전트 설치와 제거

- 3.1 에이전트 설치
  - 3.2 Tuxedo 환경에서 에이전트 적용
  - 3.3 Tmax 환경에서 에이전트 적용
  - 3.4 에이전트 기동과 종료
  - 3.5 UI 접속
  - 3.6 에이전트 제거
-

## 3. 에이전트 설치와 제거

본 장에서는 ENPharos TP 에이전트를 설치하고 제거 하는 방법에 대해 기술한다.

### 3.1. 에이전트 설치

본 장에서는 Unix 또는 Linux 운영체제에서 ENPharos TP 에이전트를 TP Monitor에 설치하는 방법을 설명한다. ENPharos TP는 인스톨러 파일을 제공한다. 인스톨러는 ENPharos TP를 처음 설치할 경우에 ENPharos TP 실행파일, 샘플, 환경변수 등 모든 요소를 설치하기 위하여 사용한다.

---

#### 참고

1. 별도의 계정을 사용하여 에이전트를 설치 할 때는 TP Monitor의 계정이 ENPharos TP 에이전트 파일들에 접근할 수 있도록, 적절한 읽기 및 실행 권한을 부여해야 한다.
  2. ENPharos TP : ENPharos TP 에이전트를 말한다.
  3. TP Monitor : ENPharos TP에서 정보를 수집할 모니터링 서버를 말한다. (Tuxedo, Tmax)
- 

#### 3.1.1. 에이전트 인스톨러 실행

에이전트를 설치하기 위해서는 에이전트 바이너리 파일인 「pharostp-AIX6.1-tuxedo-64bit-oracle-3.5.bin」 파일이 필요하다. (설치 대상 장비가 AIX6이라 가정)

1. ENPharos TP 설치 CD를 넣고 마운트 한다.

2. APM 계정의 홈 디렉터리에 「pharostp-AIX6.1-tuxedo-64bit-oracle-3.5.bin」를 복사한다.

```
pharostp@AIX6:/home/pharostp> ls  
pharostp-AIX6.1-tuxedo-64bit-oracle-3.5.bin
```

3. 콘솔에서 「pharostp-AIX6.1-tuxedo-64bit-oracle-3.5.bin」 파일을 실행 시킨다.

```
pharostp@AIX6:/home/pharostp> ./pharostp-AIX6.1-tuxedo-64bit-oracle-3.5.bin
```

콘솔에서 실행을 했을 때 아래 [그림] 3-1 과 같은 화면이 나타난다.

설치과정 선택항목에서 「1. PharosTP Install」을 선택하고, 설치할 디렉터리의 path를 입력한다.

여기에서는 「/home/pharostp/apm」 이라고 하여 pharostp 계정의 홈 디렉터리 아래에 apm이라는 디렉터리에 설치한다.

```
#####
#                                                                 #
#           ENPharosTP Install(Release)                         #
#                                                                 #
#   This Software is made by DABOMSOFT Co.,Ltd.'                #
#                                                                 #
#                               DABOMSOFT Co., Ltd.              #
#                                                                 #
#####

Do you want to install(Release) PharosTP-AIX6.1-tuxedo-64 ? [y/n] : y

o Select one of the following :

    1. ENPharosTP Install
    2. ENPharosTP Release & Patch
    3. Quit Installation Program

Selected Processing: 1

o Enter the full path of the directory to install!
  ex) /home/pharosTP/pharostp
  Input -> /home/pharosTP/apm_____
  Checking directory '/home/pharosTP/apm' :    [I] Directory does not exist.
  [I] Create PharosTP directory /home/pharosTP/apm

Now, installation is ready! Press RETURN to continue...
( Press Ctrl-C to quit )

=====[ Installation is completed successfully ]=====
ENPharosTP Installation is completed successfully.
Thank you for using ENPharosTP !

                                Technical Support Center

                                Homepage: http://www.dabomsof.com
                                E-mail  : support@dabomsoft.com

=====
```

[그림] 3-1 ENPharos TP 에이전트 설치 정보

### 3.1.2. 설치 확인

ENPharos TP 에이전트가 설치되면 다음과 같은 디렉터리가 생성된다.

```
+---- bin
+---- config
+---- data
+---- lib
+---- log
+---- path
+---- license
+---- usrinc
```

ENPharos TP 에이전트에서 사용하는 고유의 디렉터리는 다음과 같다.

디렉터리	설명
bin	ENPharos TP 에이전트의 명령어/유틸리티를 포함하는 디렉터리이다
config	환경설정 파일을 저장하는 디렉터리이다. 시스템 변수 : PHAROS_CONFIG
custom	어댑터 프로세스 중 psysrmon 바이너리를 생성하기 위한 파일을 저장하는 디렉터리이다.
data	수집서버 장애 시 데이터 백업을 위한 디렉터리이다
lib	ENPharos TP 에이전트 라이브러리 파일을 저장하는 디렉터리이다.
log	ENPharos TP Log 파일을 저장하는 디렉터리이다. 시스템 변수 : PHAROS_LOGDIR
path	프로세스 간 내부 통신을 위한 네임드 파이프(named pipe)가 생성될 디렉터리이다.
license	ENPharos TP 에이전트에 대한 License 파일을 저장하는 디렉터리이다.
usrinc	사용자 프로그램 변경 시 사용할 헤더 파일을 저장하는 디렉터리이다.

[표] 3-1 ENPharos TP 에이전트 디렉터리 설명

## 3.2. Tuxedo 환경에서 에이전트 적용

설치가 완료되면 ENPharos TP 계정의 환경파일 「.profile」에 자동으로 환경변수가 등록된다. 다음은 Bourne shell 또는 Korn shell 사용자가 ENPharos TP 에이전트를 설치하면 「.profile」에 추가되는 ENPharos TP 계정의 환경변수 보여준다. 자세한 내용은 플랫폼 마다 다를 수 있다.

### 3.2.1. ENPharos TP 관리자 계정의 환경변수 확인

```
# ===== #
#  PharosTP Environment Set                               #
# ===== #
. /home/pharostp/apm/bin/pharosadt.env
```

[그림] 3-2 Pharso TP 관리자 환경변수

「pharosadt.env」파일의 정보는 다음과 같다.

```
#=====#
# PHAROSTP APM Environment  ( apm environment set )
#=====#
export PHOMEDIR=/home/pharosTP/apm
export PHAROS_TPHOME=$PHOMEDIR/pharostp
export PHAROS_CONFIG=pharostp.cfg
export PHAROS_LOGDIR=$PHOMEDIR/log
export LOGLEVEL=EWIDTH
export OS_VENDER=ibm64
export TP_TYPE=tuxedo
export DB_VENDER-oracle
export ORACLE_VERSION=ORA64
export TP_SERVER_DOWN=false
export LIBPATH=$LIBPATH:$PHOMEDIR/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$PHOMEDIR/lib
```

[그림] 3-3 pharosadt.env 설정 내용

환경변수	내용
PHOMEDIR	ENPharos TP 에이전트의 홈 디렉터리이다.
PHAROS_TPHOME	ENPharos TP 프로젝트 홈 디렉터리이다.
PHAROS_CONFIG	ENPharos TP 에이전트의 환경 파일을 지정한다.
PHAROS_LOGDIR	ENPharos TP 에이전트의 로그 파일 위치를 지정한다.
LOGLEVEL	ENPharos TP 에이전트의 로그레벨을 지정한다. 로그레벨은 디버그 모드로 실행되는 환경에서만 적용된다.

	- E : Error / W : Warning / I : Info / D : Debug / T : Trace / H : Hexdump ● 설정예 LOGLEVEL=EWIDT
OS_VENDER	ENPharos TP 에이전트가 실행되는 시스템 벤더를 지정한다 - HP-UX: hp64 / IBM AIX: ibm64 /SUN: sun64 ● 설정예 OS_VENDER=ibm64
DB_VENDER	Tmax나 Tuxedo 서버 프로그램에서 사용하는 데이터베이스 벤더를 지정한다. - Oracoe : oracle, Altibase: altibase ● 설정예 DB_VENDER=oracle
ORACLE_VERSION	서버 프로세스에서 접속하는 데이터베이스 버전 번호를 등록한다. oracle이라고만 해도 무방하다. - Oracle 32 Bit : ORA32 / Oracle 64 Bit : ORA64 ● 설정예 ORACLE_VERSION=ORA64
TP_TYPE	TP Monitor 종류를 등록한다. - Tuxedo : tuxedo / Tmax : tmax ● 설정예 TP_TYPE=tuxedo
TP_SERVER_DOWN	서버 프로세스가 실행되면서 ENPharos TP 에이전트에 접속하지 못한 경우에 서버 프로세스를 종료할 것인지 여부를 등록한다. ● 설정예 true : 접속 오류 시, 서버 다운 false : 접속 오류 시, 서버 다운하지 않음
LIBPATH	ENPharos TP 에이전트 공유라이브러리 경로를 지정한다.
LD_LIBRARY_PATH	ENPharos TP 에이전트 공유라이브러리 경로를 지정한다.

[표] 3-2 pharosadt.env 환경 변수 설명

### 3.2.2. ENPharos TP 관리자 계정의 환경설정 추가

ENPharos TP 관리자 계정의 「.profile」에 Oracle, Tuxedo 환경설정을 추가해야 한다. ENPharos TP 에이전트에서 Oracle 클라이언트 라이브러리와 Tuxedo 라이브러리를 참조하기 때문에 경로가 설정되어 있지 않거나, 다른 ENPharos TP 에이전트 Boot 중 오류가 발생한다.

#### 3.2.2.1. ORACLE 환경 설정

```
#=====#
# Oracle Environment
#=====#
export ORACLE_BASE=/oracle/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/10.2.0
export ORACLE_SID=KFCORAP1
export NLS_LANG=American_America.K016KSC5601
export NLS_DATE_FORMAT="YYYY/MM/DD"
export PATH=$PATH:$ORACLE_HOME/bin:$ORACLE_BASE/patch/OPatch
export LIBPATH=$LIBPATH:$ORACLE_HOME/lib:$ORACLE_HOME/lib
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

[그림] 3-4 ORACLE 환경변수

#### 3.2.2.2. Tuxedo 환경 설정

```
#=====#
# Tuxedo Environment
#=====#
export TUXDIR=/home/Tuxedo
export TUXCONFIG=$TUXDIR/config/tuxconfig;
export BDMCONFIG=$TUXDIR/config/dmconfig;
export TLOGDEVICE=$TUXDIR/log/tlog/TLOG;
export ULOGPFX=$TUXDIR/log/ulog/ULOG;
export ALOGPFX=$TUXDIR/log/ulog/ACCESS;
export LIBPATH=$LIBPATH:$TUXDIR/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$TUXDIR/lib
```

[그림] 3-5 Tuxedo 환경변수

### 3.2.3. ENPharos TP 라이브러리 권한 변경

Tuxedo 관리자 계정과 ENPharos TP 에이전트 계정이 다른 경우에 ENPharos TP 에이전트에서 제공하는 라이브러리를 Tuxedo 서버 프로세스에서 참조하기 때문에 라이브러리에 대해서 실행 권한을 주어야 한다.

```
pharostp@AIX6:/home/pharostp/apm/lib> ls -al
-rwx----- 1 pharostp dba      232907 Apr 26 13:28 libpharos.so*
-rwx----- 1 pharostp dba      241867 Nov  7 15:58 libpharosdb.so*
-rwx----- 1 pharostp dba      120062 Apr 29 22:16 libpharostpcom.so*
-rwx----- 1 pharostp dba     111847 Apr 30 11:32 libpharostpdaehm.dbg.so*
-rwx----- 1 pharostp dba       73138 Apr 26 13:29 libpharostpdaehm.so*
-rwx----- 1 pharostp dba     113368 Apr 27 02:19 libpharostpshm.dbg.so*
-rwx----- 1 pharostp dba      98878 Apr 26 13:29 libpharostpshm.so*
-rwx----- 1 pharostp dba      34045 Apr 27 02:19 libpharostpsh.dbg.so*
-rwx----- 1 pharostp dba      32099 Apr 26 13:29 libpharostpsh.so*
-rwx----- 1 pharostp dba      85985 Apr 26 13:30 libpsysrmon.dbg.so*
-rwx----- 1 pharostp dba      38597 Apr 26 13:30 libpsysrmon.so*
-rwx----- 1 pharostp dba     230604 Apr 26 13:30 libtpadmin.dbg.so*
-rwx----- 1 pharostp dba      84589 Apr 26 13:30 libtpadmin.so*
-rwx----- 1 pharostp dba       1540 Apr 26 13:28 libsigusrproc.so*

pharostp@AIX6:/home/pharostp/apm/lib> chmod 755 *.so
-rwxr-xr-x 1 pharostp dba      232907 Apr 26 13:28 libpharos.so*
-rwxr-xr-x 1 pharostp dba      241867 Nov  7 15:58 libpharosdb.so*
-rwxr-xr-x 1 pharostp dba      120062 Apr 29 22:16 libpharostpcom.so*
-rwxr-xr-x 1 pharostp dba     111847 Apr 30 11:32 libpharostpdaehm.dbg.so*
-rwxr-xr-x 1 pharostp dba       73138 Apr 26 13:29 libpharostpdaehm.so*
-rwxr-xr-x 1 pharostp dba     113368 Apr 27 02:19 libpharostpshm.dbg.so*
-rwxr-xr-x 1 pharostp dba      98878 Apr 26 13:29 libpharostpshm.so*
-rwxr-xr-x 1 pharostp dba      34045 Apr 27 02:19 libpharostpsh.dbg.so*
-rwxr-xr-x 1 pharostp dba      32099 Apr 26 13:29 libpharostpsh.so*
-rwxr-xr-x 1 pharostp dba      85985 Apr 26 13:30 libpsysrmon.dbg.so*
-rwxr-xr-x 1 pharostp dba      38597 Apr 26 13:30 libpsysrmon.so*
-rwxr-xr-x 1 pharostp dba     230604 Apr 26 13:30 libtpadmin.dbg.so*
-rwxr-xr-x 1 pharostp dba      84589 Apr 26 13:30 libtpadmin.so*
-rwxr-xr-x 1 pharostp dba       1540 Apr 26 13:28 libsigusrproc.so*
```

### 3.2.4. Tuxedo 관리자 계정의 환경변수 등록

Tuxedo 관리자 계정에도 ENPharos TP 에이전트 환경을 적용해야 한다. 이는 ENPharos TP 라이브러리를 이용하여 Tuxedo의 정보를 수집하기 때문이다. 관리자 계정 「.profile」에 다음과 같이 환경변수를 추가 해 준다.

```
# 기존 Tuxedo 계정의 환경 설정들
~
. /home/pharostp/apm/bin/pharostp.env
```

[그림] 3-6 Tuxedo 계정 환경변수



「pharostp.env」 파일의 정보는 다음과 같다.

```
#=====#
# PHAROSTP TPM Environment ( TP-M environment set )
#=====#
export PHOMEDIR=/home/pharostp/apmW
export PHAROS_TPHOME=$PHOMEDIR/pharostp
export PHAROS_CONFIG=pharostp.cfg
export PHAROS_LOGDIR=$PHOMEDIR/log
export LOGLEVEL=EWIDT
export OS_VENDER=ibm64
export TP_TYPE=tuxedo
export DB_VENDER=oracle
export ORACLE_VERSION=oracle
export TP_SERVER_DOWN=false
export LIBPATH=$LIBPATH:$PHOMEDIR/lib:$LIBPATH
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$LD_LIBRARY_PATH

#=====#
# export preload (tuxedo= libpharostpsh.so, tmax=libpharosload.so)
#=====#
export PCUSTOM_PRELOAD=$PHOMEDIR/lib/libcustomhook.so
if [ $OS_VENDER = ibm64 ]; then
    export LDR_PRELOAD64=$PHOMEDIR/lib/libpharostpsh.so
elif [ $OS_VENDER = ibm32 ]; then
    export LDR_PRELOAD=$PHOMEDIR/lib/libpharostpsh.so
elif [ $OS_VENDER = sun64 ]; then
    export LD_PRELOAD_64=$PHOMEDIR/lib/libpharostpsh.so
else
    export LD_PRELOAD=$PHOMEDIR/lib/libpharostpsh.so
fi
```

[그림] 3-7 pharostp.env 설정 내용

환경변수	내 용
PRELOAD 변수	<p>공유 라이브러리 Hooking을 위한 라이브러리 명을 등록한다. PRELOAD 변수는 사용하는 TP Monitor에 따라서 등록 여부가 결정된다. Tuxedo는 PRELOAD 환경변수를 아래와 같이 등록하고, Tmax는 등록하지 않는다.</p> <ul style="list-style-type: none"> <li>● [Tuxedo] 설정예</li> </ul> <p>IBM 32 : LDR_PRELOAD32=\$PHOMEDIR/lib/libpharostpsh.so IBM 64 : LDR_PRELOAD64=\$PHOMEDIR/lib/libpharostpsh.so 기타 OS : LDR_PRELOAD=\$PHOMEDIR/lib/libpharostpsh.so</p>
PCUSTOM_PRELOAD	<p>사용자 함수와 모듈에 대한 Hooking 라이브러리를 사용 시 전체경로 및 라이브러리명을 등록한다. 기본값은 설정되어 있지 않다. 사용자 Hooking 라이브러리는 pcustomgen 유틸리티와 pcustomgen.cfg 구성 파일을 이용하여 생성한다.</p>
TP_SERVER_DOWN	<p>서버 프로세스가 실행되면서 ENPharos TP 에이전트에 접속하지 못한 경우에 서버 프로세스를 종료할 것인지 여부를 등록한다.</p> <ul style="list-style-type: none"> <li>● 설정예</li> </ul> <p>true : 접속 오류 시, 서버 다운 false : 접속 오류 시, 서버 다운하지 않음</p>

[표] 3-3 pharostp.env 환경 변수 설명

## 주의

Tuxedo 관리자 계정과 ENPharos TP 에이전트 계정이 다른 경우에 pharostp.env 파일을 Tmax 관리자 계정에서 참조하기 때문에 해당 파일에 대한 읽기 권한을 주어야 한다.

## 3.2.5. ENPharos TP 설정 파일 수정 (pharostp.cfg)

ENPharos TP 계정과 TP Monitor 시스템 계정의 환경변수가 정상적으로 등록되었다면, ENPharos TP 에이전트 환경파일을 설정해 주어야 한다. 전체 에이전트 환경파일에 대한 설명은 “ENPharos TP Administration Guide”에서 자세하게 다루고 있으므로 본 안내서에서는 실행에 필요한 설정 방법에 대해서만 설명한다.

ENPharos TP 에이전트 환경파일은 config 디렉터리 아래에서 확인할 수 있는데 pharostp.cfg 를 수정한다.

```
pharostp@AIX6:/home/pharostp/apm/config> ls
customgen.sample.cfg  pharostp.sample.cfg
pharostp@AIX6:/home/pharostp/apm/config> cp pharostp.sample.cfg pharostp.cfg
pharostp@AIX6:/home/pharostp/apm/config> ls
customgen.sample.cfg  pharostp.cfg          pharostp.sample.cfg
```

### 3.2.5.1. 에이전트 정보 등록

pharostp.cfg 를 자신의 환경에 맞게 설정한다.

```
#####
# Instance information
#####
[instance_info]

# adapter name
instance.res_name=aix6_TP

# adapter name make flag
# true : agent name = res_name + tp_username
instance.res_name_join=false

# adapter unique number
instance.res_id=100
```

[그림] 3-8 pharostp.cfg - 에이전트 정보

- instance.res\_name=string [12자 이내]  
UI상에 표시될 에이전트 이름을 등록한다. 에이전트 이름은 멀티 노드 시스템 환경인 경우 유일한 이름을 등록해야 한다.
- instance.res\_name\_join=true | false  
에이전트 하나로 두 개 이상의 TP 인스턴스를 관리하는 경우에 에이전트 명을 구분하기 위해 시스템 사용자 명을 에이전트에 추가할 것인지 여부를 등록한다. 시스템 사용자 명을 에이전트 명에 추가하는 경우에 에이전트 명이 12자리가 넘지 않도록 주의해야 한다.
- instance.res\_id=numeric  
에이전트를 구분하기 위한 유일한 번호를 등록한다.

### 3.2.5.2. TP Monitor 정보 등록

```
#####  
# TP Monitor information  
#####  
[tpmon_info]  
  
# 거래추적 대상 tp monitor name  
tpmon.tp_name=TUXEDO  
  
# 거래추적 대상 tp monitor version  
tpmon.tp_ver=11.1.1.2.0  
  
# 거래추적 대상 tp monitor 실행 계정 명  
tpmon.tp_username=pharosTP  
  
# database kind  
tpmon.db_name=oracle  
  
# database connection info  
tpmon.db_coninfo=pharosdb/pharosdb@FAUST11G  
  
# 모니터링 대상 TP Monitor Engine 프로세스  
tpmon.engine_proc=BBL, WSL, WSH, DMADM, GWADM, GWTDOMAIN  
#tpmon.engine_proc=tmm,clh,cli,tms_ora
```

[그림] 3-9 pharostp.cfg – TP Monitor 정보

- tpmon.tp\_name=string [8자 이내]  
거래추적 대상 TP Monitor 이름을 등록한다. (Tuxedo/ Tmax)
- tpmon.tp\_ver=string [8자 이내]  
거래추적 대상 TP Monitor 버전을 등록한다.
- tpmon.tp\_username=string [20자 이내]  
거래추적 대상 TP monitor 실행 계정 명을 기록한다.
- tpmon.db\_name=string [10자 이내]  
서버 프로세스에서 사용하는 데이터베이스 명을 등록한다.

- tpmon.db\_coninfo=string [60자 이내]  
데이터베이스 접속 정보를 등록한다. SQL 문장에 대한 SQL plan을 사용하는 경우에는 반드시 접속 정보를 등록해야 한다.
- tpmon.engine\_proc=string [100개 프로세스 이내]  
모니터링 대상 TP Monitor Engine 프로세스 명을 등록한다. 이렇게 등록된 Engine 프로세스는 주기적으로 확인하여 Engine 프로세스가 다운되었으면 이벤트를 발생한다.

### 3.2.5.3. 수집서버 정보 등록

```
#####
# Server connection information setting
#####
[server_info]

# Pharos Server IP-ADDR
server.address=175.118.115.24

# Pharos Server에 추적 정보를 전송할 포트번호 [default=8877]
server.send_port=44001

# session number for send by master server : [min=1, max=10]
server.send_session_num=1

# Pharos Server로부터 요청받을 포트번호 [default=8887]
server.recv_port=44001

# session number for receive by master server : [min=1, max=10]
server.recv_session_num=1
```

[그림] 3-10 pharostp.cfg – 수집서버 정보

- server.address=string [24자 이내]  
수집 서버의 IP Address 를 등록한다.
- server.address= numeric [1000 ~ MAX\_INT]  
에이전트에서 수집 서버로 데이터 전송을 위한 포트번호를 등록한다.
- server.send\_session\_num =numeric [1 ~ 10]  
에이전트에서 수집 서버로 데이터 전송을 위하여 연결할 세션 개수를 등록한다.

- server.recv\_port = numeric [1000 ~ MAX\_INT]

수집 서버에서 에이전트로 요청을 보내기 위해 사용할 포트번호를 등록한다.

※ port는 수집서버의 「pharos\_env.sh」 설정인 「PORT\_AGENT=44001」 과 동일하게 맞춰 주어야 한다.

- server.recv\_session\_num = numeric [1 ~ 10]

수집 서버에서 에이전트로 요청을 보내기 위하여 연결할 세션 개수를 등록한다.

#### 3.2.5.4. 모니터링 정보 등록

```
#####
# Monitoring information setting
#####
[monitoring]

#APM type [tp]
mon.type=tp

# APM내부 프로세스간 데이터 송수신 Message queue 키 값 [default=39829]
mon.msgkey=49969

# tracking level [0=NO TRACE | 1=SERVICE | 2=MODULE | 3=FUNCTION | 4=ATMI | 5=SQL]
mon.trace_level=5

# service threshold(millisecond) : [default=0]
mon.svc_cp=0

# threshold for active service(millisecond) : [default=10000]
mon.act_svc_cp=3000

# interval for send active service list(millisecond) : [default=5000]
mon.act_svc_int=5000

# interval for send cpu and memory info(millisecond) : [default=5000]
mon.sys_info_int=5000

# falg for data summer : [true|false]
mon.sum_flag=false

# interval for send summer data(millisecond) : [default=60000]
mon.sum_send_int=60000
```

```

# interval for send system info server process of tp monitor(millisecond) : [default=60000]
mon.proc_info_int=5000

# flag for system usage(CPU & Memory leak) : [true|false]
mon.svc_sys_usage=true

# flag for system info server process of tp monitor : [true|false]
mon.proc_info_flag=true

# interval for request info(millisecond) : [default=1000]
mon.req_send_int=1000

# flag for connection admin info of tp monitor : [true|false]
mon.adm_info_flag=true

# interval for send configuration info of tp monitor(millisecond) : [default=600000]
mon.adm_cfg_info_int=600000

# interval for send real time info of tp monitor(millisecond) : [default=5000]
mon.adm_rt_info_int=5000

# buffering count for less then threshold : [min=1, max=500]
mon.merge_cnt=10

# flag for buffering less then threshold by error service : [true|false]
mon.err_buf_flag=true

# 거래추적에서 제외할 서버 프로세스 명 : comma로 구분
mon.skip_svr=BBL, DBBL, LMS, WSL, WSH, DMADM, GWADM, GWTDOMAIN, tmboot, tmshutdown,
tmadmin, JSL, JREPSVR, tmunloadcf, tlisten, tuxadm, apaddusr, tmipcrm, tpmkill, tms_ora

# service name for tracking exemption : comma separator
mon.skip_svc=

# flag for tracking exemption by transaction code : [positive|negative|false]
mon.txcode_flag=false

# file name for transaction tracking by transaction code
mon.txcode_path=

```

```

# APM 헤더 처리 방법 [0=미사용|1=APM 헤더]
mon.apm_header_flag=0

# service name for skip used apm header : comma separator
mon.apm_skip_svc=

# flag for transaction tracking by hashcode : [0=Not used | 1=Message | 2=GUID]
mon.hashcode_type=0

# flag for merge one calltree by async service : [true|false]
mon.async_to_tx=false

# flag for async to sync transformation : [true|false]
# mon.async_timeout : millisecond
mon.async_to_sync=false
mon.async_start_svc=
mon.async_end_svc=
mon.async_timeout=0

# config file name for daemon process hooking
mon.proc_config=

# skip count for memory leak check : [default=10]
mon.leak_skip_cnt=10

# flag for polling data send : [true|false]
mon.polling_flag=true
# interval for polling data send(millisecond) : [default=30000]
mon.polling_int=30000

```

[그림] 3-11 pharostp.cfg – 모니터링 정보

- mon.type= tp|db|mc|link [10자 이내]  
에이전트 타입을 등록한다.
- mon.msgkey=numeric [1 ~ MAX\_INT]  
서버 프로세스의 성능정보를 에이전트에 전송하기 위해 APM내부 프로세스간 성능 정보를 송신 하기 위하여 사용할 메시지 큐의 키를 등록한다.
- mon.trace\_level=numeric [0 ~ 5]



성능 정보 추출 레벨을 등록한다. 대상이 서비스 단위인지 아니면 SQL 단위까지 성능정보를 추출 할 것인지 등 대상의 상세 레벨을 등록한다.

- `mon.svc_cp=numeric` [0 ~ 100000]

대상 서비스의 임계치를 밀리초 단위로 등록한다. 서비스 처리시간이 임계치 보다 오래 걸린 서비스에 대해서만 수집서버에 전송하여 네트워크 부하를 줄이기 위한 목적이다.

- `mon.act_svc_cp=numeric` [0 ~ 100000]

모든 Active 서비스 리스트를 수집서버에 전송하지 않고 임계치가 경과한 서비스만 수집서버에 전달하기 위한 임계치를 등록한다.

- `mon.act_svc_int=numeric` [1000 ~ 100000]

현재 수행중인 Active 서비스 리스트를 어느 정도 간격으로 수집 서버에 전달할 것인지를 밀리초 단위로 등록한다.

- `mon.sys_info_int=numeric` [1000 ~ 500000]

업무 서버의 시스템 정보(CPU, 메모리)를 어느 정도 간격으로 수집 서버에 전달할 것인지를 밀리초 단위로 등록한다.

- `mon.sum_flag=true | false`

에이전트 내부적으로 집계를 처리 할 것인지 여부를 등록한다. 최초에는 에이전트에서 통계를 생성했기 때문에 의미가 있었지만 현재는 수집서버에서 통계를 생성하기 때문에 의미가 없는 항목이므로 false로 등록한다.

- `mon.sum_send_int=numeric` [60000 ~ 1000000]

에이전트 내부적으로 집계한 집계정보를 어느 정도 간격으로 수집 서버에 전달할 것인지를 밀리초 단위로 등록한다. `sum_flag`가 false이면 동 항목은 사용하지 않는다.

- `mon.svc_sys_usage=true | false`

서비스에 대한 성능 정보 중 서비스를 처리할 때 사용한 CPU 사용량 및 메모리 Leak 정보를 수집할 것인지 여부를 등록한다.

- `mon.proc_info_flag=true | false`

TP Monitor 하에서 실행되는 서버 프로세스에 대한 CPU 및 메모리 정보를 수집할 것인지 여부를 등록한다.

- `mon.proc_info_int=numeric` [5000 ~ 100000]

TP Monitor 하에서 실행되는 서버 프로세스에 대한 CPU 및 메모리 정보를 어느 정도 간격으로 수집 서버에 전달할 것인지를 밀리초 단위로 등록한다. 정보를 수집하는 경우 "tpmon.tp\_username" 항목의 계정으로 실행된 프로세스에 대한 정보만 수집한다.

- `mon.req_send_int=numeric` [100 ~ 100000]  
입력된 거래 건수와 종료된 거래 건수 정보를 어느 정도 간격으로 수집 서버에 전달할 것인지를 밀리초 단위로 등록한다.
- `mon.adm_info_flag=true | false`  
TP Monitor Admin 정보를 수집할 것인지 여부를 등록한다. Admin 정보로는 서버 프로세스, 서비스 Queue 정보 등이 있다.
- `mon.adm_cfg_info_int=numeric` [300000 ~ 1000000]  
TP Monitor Admin 정보 중 Config 성 정보를 어느 정도 간격으로 수집서버에 전달할 것인지를 밀리초 단위로 등록한다.
- `mon.adm_rt_info_int=numeric` [1000 ~ 100000]  
TP Monitor Admin 정보 중 실시간성 정보를 어느 정도 간격으로 수집서버에 전달할 것인지를 밀리초 단위로 정의한다.
- `mon.merge_cnt=numeric` [1 ~ 500]  
임계치 이하 서비스에 대한 집계정보를 매번 수집 서버에 전송하면 부하가 발생하므로, 이를 최소화하기 위하여 버퍼링하여 수집 서버에 전송한다. 이를 위한 버퍼링 건수를 등록한다.
- `mon.err_buf_flag=true | false`  
임계치 이하 데이터 중 에러가 발생한 서비스에 대해서 버퍼링 할 것인지 아니면 Call-Tree 정보를 전송할 것인지 여부를 등록한다.
- `mon.skip_svr=string`  
거래추적에서 제외할 서버 프로세스 명을 등록한다. (윗 라인이 Tuxedo, 아래 라인이 Tmax)
- `mon.skip_svc=string`  
성능정보 추출에서 제외할 서비스 명을 등록한다. 등록할 서비스가 두 개 이상인 경우 Comma로 분리한다. 특수문자 '\*' 문자를 이용하여 등록이 가능하다. 특수문자 '\*' 문자를 이용하여 등록하는 경우에 '\*' 문자를 처음이나 중간에 사용할 수는 없고 항상 뒤에 사용해야 한다.
- `mon.tx_code_flag=positive | negative | false`  
등록된 거래코드에 대해서 성능정보를 추출 할 것인지 아니면 제외할 것인지, 또는 거래코드를 등록하지 않을 것인지 여부를 등록한다.
- `mon.txcode_path=string`  
거래코드 리스트가 등록된 파일명을 등록한다. `tx_code_flag` 항목이 false가 아니면 동 항목에 거래코드가 등록된 파일명을 등록해야 한다.

- `mon.apm_header_flag=numeric` [0 ~ 1]  
거래추적을 위해서 APM 헤더를 사용할 것인지 여부를 등록한다. APM 헤더를 사용하면 서비스를 호출할 때 자동으로 APM 헤더를 삽입하여 호출한다.
- `mon.apm_skip_svc=string`  
거래추적을 위해 APM 헤더를 사용할 때 하나의 TP 시스템 속하는 모든 서비스는 기본적으로 거래추적을 위하여 서비스간 호출 시 GUID 정보 전달을 위해서 APM 헤더를 전달하는데, 이때 거래추적에서 제외할 서비스가 있는 경우 APM 헤더를 전달하지 않기 위하여 예외 서비스를 등록하는 항목이다
- `mon.hashcode_type=numeric` [0 ~ 2]  
전문의 GUID가 있는 경우 APM 헤더를 사용하지 않고 거래를 추적할 수 있다. 이때 거래를 하나의 Call-Tree로 만들기 위해서는 Caller와 Callee간 연결번호가 있어야 하는데, 해당 번호를 무엇으로 할 것 인지를 등록한다.
- `mon.async_to_tx=true | false`  
비동기 호출(tpacall)에 대해서도 하나의 거래로 인식하여 Call-Tree로 처리할 것인지 여부를 등록한다.
- `mon.async_to_sync=true | false`  
비동기 거래를 동기 거래로 변환하여 처리할 것인지 여부를 등록한다.
- `mon.async_start_svc=string`  
"mon.async\_to\_sync" 항목이 true인 경우에 비동기 거래의 시작 서비스 명을 등록한다.
- `mon.async_end_svc=string`  
"mon.async\_to\_sync" 항목이 true인 경우에 비동기 거래의 종료 서비스 명을 등록한다.
- `mon.async_timeout=numeric`  
"mon.async\_to\_sync" 항목이 true인 경우에 비동기 거래의 타임아웃 값을 등록한다.
- `mon.proc_config=string`  
TP Monitor 환경에서 TP에 등록된 서버 프로세스 이 외에 데몬 형식으로 실행되는 프로세스에 대해 성능 정보를 수집하기 위해서 데몬 프로세스의 정보가 등록된 파일명을 등록한다.
- `mon.leak_skip_cnt=numeric` [10 ~ MAX-INT]  
`svc_sys_usage` 항목이 true이면 서비스를 처리할 때 사용한 메모리 량을 계산하여 메모리 Lack이 발생했는지 체크하게 되는데 처음 프로세스가 실행 될 때부터 체크하게 되면 거의 대부분 메모리 Leak이 발생하게 된다. 이런 문제점을 해결하기 위해 처음 몇 번 동안에는 메모리 Leak을 조사하지 않도록 제외할 횟수를 등록하는 항목이다.

- mon.polling\_flag=true | false

업무서버와 수집서버 사이에 방화벽이 있는 경우에 일정 시간 동안 데이터 통신이 없으면 자동으로 방화벽에서 세션을 끊기 때문에 수집서버와 통신할 때 장애가 발생한다. 이를 위해 Polling 데이터를 주기적으로 전송할 것인지 여부를 등록한다.

- Mon.polling\_int=numeric [10000 ~ 600000]

polling\_flag 항목이 true인 경우에 어느 정도 간격으로 Polling 데이터를 전송할 것인지 밀리초 단위로 등록한다.

### 3.2.5.5. TUXEDO 어드민 모니터링 정보

```
#####
# TUXEDO Admin Monitoring information setting
#####
[tpadmin]

# Queue Data Query Mode (1=tpadmin -r, 2=tpadmin -r file name)
tpadmin.queue_inquiry=

# Queue File Full Path
tpadmin.queue_file=

# tuxedo LMID (2014/08/04 add)
tpadmin.tux_lmid=TUXINT_AP1
```

[그림] 3-12 pharostp.cfg – TUXEDO 어드민 모니터링 정보

- tpadmin.queue\_inquiry=numeric [1 ~ 2]

Tuxedo도의 경우 큐 정보를 조회할 때 Tuxedo admin 명령어를 사용하여 큐 정보를 구하게 되는데, Tuxedo admin 명령어의 권한이 없어 실행하지 못하는 문제가 있다. 이를 위해 Tuxedo 계정에서 admin 명령어로 큐 정보를 파일로 만들어 주면 동 파일을 읽어서 처리할 수 있다. 동 항목은 Tuxedo admin명령어를 직접 사용할 것인지 아니면 파일에서 큐 정보를 조회할 것인지 여부를 등록한다.

- tpadmin.queue\_file=string

queue\_inquiry 항목의 값이 "2"인 경우에 파일을 통해서 큐 정보를 조회하는데, 큐 정보가 저장된 파일명을 등록한다.

- tpadmin.tux\_lmid=string

ENPharos TP 에이전트가 설치되어 있는 Tuxedo 시스템에 대한 큐 정보를 실시간으로 알려주기 위해 사용하는 항목으로 Tuxedo 구성파일 MACHINES 에 설정되어 있는 LMID의 값을 등록한다. 최대 16자 이내로 등록한다.

### 3.2.5.6. 모듈 모니터링 정보

```
#####  
# Module Monitoring information setting  
#####  
[module]  
# 모듈 임계치(밀리초) : 임계치 이하 데이터는 Call-Tree에 나타나지 않음 [default=0]  
# 단, 하위에 Entry가 존재시 임계치 이하 데이터도 포함  
module.mod time limit=0
```

[그림] 3-13 pharostp.cfg – 모듈 모니터링 정보

- module.mod\_time\_limit=numeric [0 ~ MAX\_INT]

모듈 처리 시간에 대한 임계치를 등록하는 항목이다. 동 항목을 설정하면 임계치 이상의 모듈에 대해서만 Call-Tree에 표현한다. 단, 해당 모듈 하위에 Call-Tree가 존재하는 경우에는 임계치에 관계없이 Call-Tree에 표현한다.

### 3.2.5.7. SQL 모니터링 정보

```
#####  
# SQL Monitoring information setting  
#####  
[sql_monitoring]  
  
# Session 정보 수집 여부 [true|false]  
sql.sid_coll_flag=true  
  
# SQL bind data 수집 여부 [true|false]  
sql.bind_coll_flag=true  
  
# SQL plan 여부 [true|false]  
sql.sql_plan_flag=true  
  
# SQL TEXT hashing 여부 [true|false]  
sql.sql_hash_flag=true  
  
# SQL 임계치 예외 건수: 임계치와 관계없이 call tree에 포함할 개수 [default=100]  
sql.sql_stack_cnt=100
```

```
# SQL 임계치(밀리초) : 임계치 이하 데이터는 call tree에 나타나지 않음 [default=0]
sql.sql_time_limit=0

# code list for exception sql error code : [max 128 number]
sql.skip_err_code=

# commit/rollback flag [default=false]
sql.tran_end_flag=false
```

[그림] 3-14 pharostp.cfg – SQL 모니터링 정보

- sql.sid\_coll\_flag=true | false  
서버 프로세스에서 데이터베이스 연결 정보인 세션 정보를 조회할 것인지 여부를 등록한다.
- sql.bind\_coll\_flag=true | false  
SQL 텍스트 중 바인드 데이터를 구할 것인지 여부를 등록한다.
- sql.sql\_plan\_flag=true | false  
SQL 텍스트에 대해서 Query Plan 실행할 것인지 여부를 등록한다.

## 참고

SQL Plan를 실행하기 위해서는 반드시 "3.2.5.2. TP Monitor 정보" 항목 중 "db\_coninfo" 항목을 등록해야 한다. 또한 Plan을 실행하는 모든 테이블에 대해서 권한을 부여해야 한다.

- sql.sql\_hash\_flag=true | false  
SQL 문장에 대해서 에이전트 내부적으로 해싱하여 SQL 문장을 저장하고, 동일 SQL 문장에 대해서는 해쉬코드만 수집서버에 전달하는 기능을 사용할 것인지 여부를 등록한다. 이렇게 함으로써 수집서버에 전송하는 데이터량을 상당히 줄일 수 있다.
- sql.sql\_stack\_cnt=numeric [1 ~ 1020]  
SQL 처리 시간에 대한 임계치가 설정되어 있는 경우 SQL 임계치 이하의 데이터는 Call-Tree에 표현하지 않는다. 그러나 모든 SQL이 SQL 임계치 이하이면 Call-Tree에 아무것도 표현하지 않기 때문에, 이것을 방지하기 위하여 기본적으로 나타나야 할 SQL Call-Tree 건수를 등록하는 항목이다. 동 항목에 등록된 개수 만큼은 SQL 임계치에 관계없이 Call-Tree에 표현한다.
- sql.sql\_time\_limit=numeric [0 ~ MAX\_INT]  
SQL 처리 시간에 대한 임계치를 등록하는 항목이다. 동 항목을 설정하면 위의 「sql.sql\_stack\_cnt」 항목 값에 설정된 건수에 추가적으로 임계치 이상의 SQL에 대해서만 Call-Tree만 표현한다. 즉, 「sql.sql\_stack\_cnt」 항목을 초과한 경우에는 동 항목에 정의된 SQL 임계치 이상의 SQL에 대해서만 Call-Tree에 포함한다.

- sql.skip\_err\_code=string  
SQL 처리 시 에러가 발생하는 경우 해당 에러를 에러로 인식하지 않고 정상으로 처리하고자 하는 경우 예외 에러코드를 등록한다.
- sql.tran\_end\_flag=true|false  
Commit/rollback SQL에 대하여 sql.sql\_time\_limit에서 설정한 임계 값에 따라 Call-Tree에 표현을 할지, 안 할지 여부를 등록한다.

### 3.2.5.8. 포지션 정보

```
#####
# Monitor information setting
#####
[position]

# Buffer type이 CARRAY, STRING인 경우 업무코드 위치 및 길이 (위치는 1부터 시작)
#position.appl_code_pos=96
#position.appl_code_len=4

# Buffer type이 CARRAY, STRING인 경우 거래코드 위치 및 길이 (위치는 1부터 시작)
#position.svc_name_flag=true
#position.txcode_pos=100
#position.txcode_len=10

# Buffer type이 CARRAY, STRING인 경우 글로벌ID 위치 및 길이 (위치는 1부터 시작)
#position.gid_pos=10
#position.gid_len=30

# Buffer type이 CARRAY, STRING인 경우 글로벌ID 일련번호 위치 및 길이 (위치는 1부터 시작)
#position.gid_seq_pos=40
#position.gid_seq_len=2

# Buffer type이 CARRAY, STRING인 경우 사용자 메시지 위치 및 길이 (위치는 1부터 시작)
#position.user_msg_pos=10
#position.user_msg_len=30
```

[그림] 3-15 pharostp.cfg – 포지션 정보

- position.appl\_code\_pos=numeric  
버퍼 타입이 CARRAY나 STRING인 경우 전문 내에서 업무코드 시작 위치를 등록한다.

- position.appl\_code\_len=numeric  
버퍼 타입이 CARRAY나 STRING인 경우 전문 내에서 업무코드 길이를 등록한다.
- position.svc\_name\_flag=true | false  
거래코드가 서비스 명과 동일한지 여부를 등록한다.  
true : 서비스 명과 거래코드가 동일  
false : 서비스 명과 거래코드가 동일하지 않음
- position.txcode\_pos=numeric [1 ~ 12]  
버퍼 타입이 CARRAY나 STRING인 경우 전문 내에서 거래코드 시작 위치를 등록한다.
- position.txcode\_len=numeric  
버퍼 타입이 CARRAY나 STRING인 경우 전문 내에서 거래코드 길이를 등록한다.
- position.gid\_pos = numeric  
버퍼 타입이 CARRAY나 STRING인 경우 전문 내에서 글로벌 ID 시작 위치를 등록한다.
- position.gid\_len = numeric  
버퍼 타입이 CARRAY나 STRING인 경우 전문 내에서 글로벌 ID의 길이를 등록한다.
- position.gid\_seq\_pos = numeric  
버퍼 타입이 CARRAY나 STRING인 경우 전문 내에서 글로벌 ID 일련번호 시작 위치를 등록한다. 현재는 사용하지 않기 때문에 등록하지 않아도 된다.
- position.gid\_seq\_len = numeric  
버퍼 타입이 CARRAY나 STRING인 경우 전문 내에서 글로벌 ID의 일련번호 길이를 등록한다. 현재는 사용하지 않기 때문에 등록하지 않아도 된다.
- position.user\_msg\_pos = numeric  
버퍼 타입이 CARRAY나 STRING인 경우 전문 데이터의 일부를 UI에 표현할 수 있다. UI에 표현할 전문 데이터의 시작 위치를 등록한다.
- position.user\_msg\_len = numeric [1 ~ 300]  
버퍼 타입이 CARRAY나 STRING인 경우 UI에 표현할 전문 데이터의 길이를 등록한다.



### 3.2.5.9. 에러위치 정보

```
#####  
# Application error information setting  
#####  
[app_errinfo]  
# 에러전문 판단정보. 메시지판단 위치정보, 메시지판단 코드길이, 에러전문코드 값  
# 위치정보는 1부터 시작  
#errinfo.errmsg_pos=221  
#errinfo.errmsg_len=1  
#errinfo.errmsg_code=2  
  
# 에러코드 위치 정보 (위치는 1부터 시작)  
#errinfo.code_pos=771  
#errinfo.code_len=10  
  
# 에러메시지 위치 정보 (위치는 1부터 시작)  
#errinfo.msg_pos=781  
#errinfo.msg_len=40
```

[그림] 3-16 pharostp.cfg – 에러위치 정보

- `errinfo.errmsg_pos=numeric`  
버퍼 타입이 CARRAY나 STRING인 경우 응답 전문 내에서 에러 유무를 판단하기 위한 코드의 시작 위치를 등록한다.
- `errinfo.errmsg_len=numeric [1 ~ 10]`  
버퍼 타입이 CARRAY나 STRING인 경우 응답 전문 내에서 에러 유무를 판단하기 위한 코드의 길이를 등록한다.
- `errinfo.errmsg_code=numeric`  
버퍼 타입이 CARRAY나 STRING인 경우 응답 전문 내에서 에러 유무를 판단하기 위한 에러코드를 등록한다. 위의 항목에서 정의한 코드가 동 항목에 등록된 코드와 일치하면 에러로 판단한다.
- `errinfo.code_pos=numeric`  
버퍼 타입이 CARRAY나 STRING인 경우 응답 전문 내에서 에러코드 시작 위치를 등록한다.
- `errinfo.code_len=numeric [1 ~ 12]`  
버퍼 타입이 CARRAY나 STRING인 경우 응답 전문 내에서 에러코드 길이를 등록한다.
- `errinfo.msg_pos=numeric`

버퍼 타입이 CARRAY나 STRING인 경우 응답 전문 내에서 에러 메시지 시작 위치를 등록한다.

- `errinfo.msg_len=numeric` [1 ~ 128]

버퍼 타입이 CARRAY나 STRING인 경우 응답 전문 내에서 에러 메시지 길이를 등록한다.

### 3.2.5.10. 로그 정보

```
#####
# Trace Logging information setting
#####
[log_info]

# 서비스 종료 로그 출력 여부
log.log_flag=false

# Hooking 에러로그 로깅 path
#log.log_path=/home/PharosTP/apm/log

# TP Monitor sys log file name
#log.sys_log_path=/home/tmax/tmax5/log/slog/slog.$(MMDDYYYY)

# |=or, &=and, !=not ex) ERROR|WARN|!tms
log.sys_log_pattern=CLH

# interval for log file check (millisecond)
log.sys_log_interval=5000

# Core File info
log.core_file_flag=false
log.core_file_path=/home/PharosTP/apm/log/corelog
```

[그림] 3-17 pharostp.cfg – 로그 정보

- `log.log_flag=true | false`  
서비스 처리가 종료된 경우 종료 정보를 에이전트 로그에 남길 것인지 여부를 등록한다.
- `log.log_path=string` [1 ~ 126]  
Hooking 모듈에서 디버그 로그를 저장하기 위한 path를 등록한다. 단순한 디버그 용으로 사용한다.
- `log.sys_log_path=string` [1 ~ 126]  
TP Monitor 시스템 로그 파일에 대한 Full Path를 등록한다. 동 항목을 등록하면 시스템 로그를 주기적

으로 조사하여 로그 패턴 항목 「log.sys\_log\_pattern」에 정의되어 있는 정보에 따라서 해당 로그 정보를 UI에 표현한다.

- log.sys\_log\_pattern= string [1 ~ 1020]  
TP Monitor 시스템 로그 파일에 대한 Tail 기능 처리시 조건에 합당한 로그만 출력할 수 있도록 패턴 정보를 등록한다 패턴을 등록하지 않으면 모든 로그를 수집서버에 전송한다.
- log.sys\_log\_interval=numeric  
TP Monitor 시스템 로그 파일 Tail 간격을 밀리초 단위로 등록한다.
- log.core\_file\_flag=true | false  
서버 프로세스가 비정상적으로 종료되는 경우에 CORE 파일을 지정된 디렉터리로 옮길 것인지 여부를 등록한다. 「true」로 등록하면 반드시 「log.core\_file\_path」 항목을 등록해야 한다.
- log.core\_file\_path=string [1 ~ 126]  
CORE 파일이 생성될 위치를 등록한다.

### 3.2.5.11. 공유메모리 정보 등록

```
#####  
# Monitoring Global Area information setting  
#####  
[mga_info]  
  
# APM 정보 저장을 위한 공유메모리 키 값 [default=78437]  
mga.shmkey=78847  
  
# process max number for tracking : [default=512]  
mga.svr_process=4096  
  
# calltree max number : [default=1024]  
mga.call_seq=512  
  
# max number for sql text hashing : [default=10000]  
mga.sql_hash_num=100000
```

[그림] 3-18 pharostp.cfg – 공유메모리 정보

- mga.shmkey=numeric [1 ~ MAX\_INT]  
서비스에 추출한 성능정보 저장을 위한 공유메모리 키 값을 기록한다.  
※ 단, 시스템 내부의 기존에 사용되는 공유 메모리 키 값과 겹쳐서는 안 된다.

- mga.svr\_process=numeric [512 ~ 10000]

TP Monitor 환경에서 실행되는 서버 프로세스 최대 수를 등록한다. TP Monitor 환경 파일에 정의된 서버 프로세스의 MAX 항목 값의 합보다 커야 한다.

- mga.call\_seq=numeric [256 ~ 1024]

하나의 서비스에서 내부적으로 호출할 모듈, ATMI call, SQL 문 등의 최대 Call-Tree 개수를 등록한다.

- mga.sql\_hash\_num=numeric [10000 ~ 100000]

SQL 문의 해쉬코드를 에이전트 내부적으로 몇 개까지 저장할 것인지 건수를 등록한다.

### 3.2.5.12. 연계서비스 정보 등록

```
#####
# Monitoring System Link Service information setting
#####
[link_service]

# 시스템 연계 서비스
linksvc.link_svc=string
```

[그림] 3-19 pharostp.cfg – 연계서비스 정보

- linksvc.link\_svc=string

타 시스템과 연계 시 호출되는 서비스 명을 등록한다.

### 3.2.5.13. 거래코드 정보 등록

```
#####
# Transaction code for skip list
#####
[code_list]
txcode=ABDC1
txcode=ABDC2
txcode=ABDC3
txcode=ABDC4
txcode=ABDC5
```

[그림] 3-20 txcode.cfg – 거래코드 파일 정보

- tx\_code=string

전문 상에서 추출할 수 있는 거래코드를 등록한다.

---

## 참고

동 항목을 등록한 경우에는 반드시 전문 내에서 거래코드를 구할 수 있는 방법을 등록해야 한다. TP 버퍼 타입이 STRING이나 CARRAY인 경우 [position] 절의 txcode\_pos, txcode\_len 항목의 값을 반드시 등록해야 한다.

---

### 3.2.5.14. 데몬 프로세스 정보 등록

```
#####
# TP DAEMON SERVER process config information file
#####
[Dpush]
alias=Dpush
txcode_pos=1287
txcode_len=8
tcp_rcv_ip=175.118.115.24
tcp_rcv_port=30004
tcp_send_ip=175.118.115.24
tcp_send_port=30003
recv_msgq_sekey=0x30000200-0x30000228
recv_msgq_key=0x3000005e
send_msgq_sekey=0x31000001-0x3100270f
send_msgq_key=0x3000006e
mq_send_file=sname
mq_rcv_file=rname
follow_process=mciclient
command_arg=Dpush -i 175.118.115.24 -p 30003
```

[그림] 3-21 데몬 프로세스 파일 정보

위의 파일에서 프로세스마다 모든 정보를 등록하는 것이 아니라 해당 프로세스에서 사용하는 항목만 등록한다.

● alias : 데몬 프로세스 명 대신에 별명을 등록하고자 하는 경우에 별명을 등록한다. 동 항목을 등록하면 UI 상의 Call-Tree에 해당 명이 서버명과 서비스 명으로 표현된다. 동 항목은 동일한 데몬 프로세스가 여러 개인 경우에 이를 구분하기 위해서 사용한다.

● txcode\_pos : 전문 내에서 거래코드 위치를 등록한다. 추출할 거래코드가 없는 경우에는 등록하지 않는다.

● txcode\_len : 거래코드 길이를 등록한다. 추출할 거래코드가 없는 경우에는 등록하지 않는다.

- **tcp\_rcv\_ip** : 데몬 프로세스가 TCP로 메시지를 수신 받는 경우에 상대방 프로세스가 실행되는 서버의 IP 주소를 등록한다. TCP로 수신 받는 프로세스가 아닌 경우에는 등록하지 않는다.
- **tcp\_rcv\_port** : 데몬 프로세스가 TCP로 메시지를 수신 받는 경우에 상대방 프로세스와 연결할 포트번호를 등록한다. TCP로 수신 받는 프로세스가 아닌 경우에는 등록하지 않는다.
- **tcp\_send\_ip** : 데몬 프로세스가 상대방 프로세스에게 TCP로 메시지를 송신하는 경우에 상대방 프로세스가 실행되는 서버의 IP 주소를 등록한다. TCP로 송신하지 않는 경우에는 등록하지 않는다.
- **tcp\_send\_port** : 데몬 프로세스가 TCP로 메시지를 송신하는 경우에 상대방 프로세스와 연결할 포트번호를 등록한다. TCP로 송신하지 않는 경우에는 등록하지 않는다.
- **recv\_msgq\_sekey** : 데몬 프로세스가 메시지 큐로부터 메시지를 읽는 경우에 메시지 큐의 키를 등록한다. 동 항목은 메시지 큐가 연속으로 여러 개인 경우에 시작번호와 끝번호를 등록한다. 등록하는 키 값을 16진수 형식으로 등록해야 한다.
- **recv\_msgq\_key** : 데몬 프로세스가 메시지 큐로부터 메시지를 읽는 경우에 메시지 큐의 키를 등록한다. 동 항목은 메시지 큐가 연속되지 않고 하나씩인 경우에 등록한다. 등록하는 키 값을 16진수 형식으로 등록해야 한다.
- **send\_msgq\_sekey** : 데몬 프로세스가 메시지 큐에 메시지를 저장하는 경우에 메시지 큐의 키를 등록한다. 동 항목은 메시지 큐가 연속으로 여러 개인 경우에 시작번호와 끝번호를 등록한다. 등록하는 키 값을 16진수 형식으로 등록해야 한다.
- **send\_msgq\_key** : 데몬 프로세스가 메시지 큐에 메시지를 저장하는 경우에 메시지 큐의 키를 등록한다. 동 항목은 메시지 큐가 연속되지 않고 하나씩인 경우에 등록한다. 등록하는 키 값을 16진수 형식으로 등록해야 한다.
- **mq\_send\_file** : 데몬 프로세스가 mq\_send 명령어를 이용하여 큐에 메시지를 저장하는 경우에 큐의 명을 등록한다. 큐명이 여러 개인 경우에 Comma로 구분한다.
- **mq\_rcv\_file** : 데몬 프로세스가 mq\_receive 명령어를 이용하여 큐로부터 메시지를 읽는 경우에 큐의 명을 등록한다. 큐명이 여러 개인 경우에 Comma로 구분한다.
- **follow\_process** : 데몬 프로세스가 메시지를 전달할 프로세스의 서비스 명을 등록한다. 상대방 프로세스가 데몬 프로세스인 경우에는 「alias」 항목에 등록된 명칭을 등록한다. 동 항목에 등록된 명칭은 Call-Tree 상의 호출 서비스 명으로 표현된다.
- **command\_arg** : 동일한 이름의 데몬 프로세스가 여러 개인 경우에 이를 구분할 수 있는 Command 아규먼트를 등록한다.

예를 들어 메시지 큐를 읽어서 TCP로 전송하는 데몬 프로세스인 경우에는 아래와 같이 등록한다.

```
[Dpush]
alias=Dpush
txcode_pos=1287
txcode_len=8
tcp_send_ip=175.118.115.24
tcp_send_port=30003
recv_msgq_sekey=0x30000200-0x30000228
recv_msgq_key=0x3000005e
follow_process=mciclient
command_arg=-i 175.118.115.24 -p 30003
```

### 3.2.6. psysrmon 생성

psysrmon은 시스템에 대한 CPU와 메모리 정보를 조회하는 프로세스이다. 그러나 시스템에 대한 CPU와 메모리 정보를 시스템 벤더에 따라서 다양한 API를 제공한다. 또한 동일한 API라도 API에서 제공되는 구조체가 다양하기 때문에 해당 시스템에 맞게 compile 되어야 한다.

ENPharos TP 에이전트에서 제공하는 라이브러리 「psysrmon.so」와 「psysrmon.c」 소스를 컴파일하여 링크하면 psysrmon 실행 파일이 생성한다.

※ ENPharos TP와 Tuxedo가 동일한 관리자 계정에 설치 되어 있다면 pharostp.env 파일만 환경변수에 등록해 주도록 한다.

## 3.3. Tmax 환경에서 에이전트 적용

설치가 완료되면 ENPharos TP 에이전트의 환경파일 「.profile」에 자동으로 환경변수가 등록된다. 다음은 Bourne shell 또는 Korn shell 사용자가 ENPharos TP 에이전트를 설치하면 「.profile」에 추가되는 ENPharos TP 에이전트 환경변수 보여준다.

### 3.3.1. ENPharos TP 관리자 계정의 환경변수 확인

```
# ===== #
#  PharosTP Environment Set                      #
# ===== #
. /home/pharostp/apm/bin/pharosadt.env
```

[그림] 3-22 ENPharos TP 관리자 환경변수

「pharosadt.env」 파일의 정보는 다음과 같다.

```

export PHOMEDIR=/home/pharosTP/apm
export PHAROS_TPHOME=$PHOMEDIR/pharostp
export PHAROS_CONFIG=pharostp.cfg
export PHAROS_LOGDIR=$PHOMEDIR/log
export LOGLEVEL=EWIDTH
export OS_VENDER=ibm64
export TP_TYPE=tmax
export DB_VENDER=oracle
export ORACLE_VERSION=oracle
export TP_SERVER_DOWN=false
export LIBPATH=$LIBPATH:$PHOMEDIR/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$PHOMEDIR/lib

```

[그림] 3-23 pharosadt.env 정보

환경변수	내용
PHOMEDIR	ENPharos TP 에이전트의 홈 디렉터리이다.
PHAROS_TPHOME	ENPharos TP 프로젝트 홈 디렉터리이다.
PHAROS_CONFIG	ENPharos TP 에이전트의 환경 파일을 지정한다.
PHAROS_LOGDIR	ENPharos TP 에이전트의 로그 파일 위치를 지정한다.
LOGLEVEL	<p>ENPharos TP 에이전트의 로그레벨을 지정한다.</p> <p>로그레벨은 디버그 모드로 실행되는 환경에서만 적용된다.</p> <p>- E : Error / W : Warning / I : Info / D : Debug / T : Trace / H : Hexdump</p> <p>● 설정예</p> <p>LOGLEVEL=EWIDT</p>
OS_VENDER	<p>ENPharos TP 에이전트가 실행되는 시스템 벤더를 지정한다</p> <p>- HP-UX: hp64 / IBM AIX: ibm64 /SUN: sun64</p> <p>● 설정예</p> <p>OS_VENDER=ibm64</p>
ORACLE_VERSION	<p>서버 프로세스에서 접속하는 데이터베이스 버전 번호를 등록한다.</p> <p>oracle이라고만 해도 무방하다.</p> <p>- Oracle 32 Bit : ORA32 / Oracle 64 Bit : ORA64</p> <p>● 설정예</p> <p>ORACLE_VERSION=ORA64</p>
TP_TYPE	<p>TP Monitor 종류를 등록한다.</p> <p>- Tuxedo : tuxedo / Tmax : tmax</p> <p>● 설정예</p> <p>TP_TYPE=tuxedo</p>
DB_VENDER	<p>Tmax나 Tuxedo 서버 프로그램에서 사용하는 데이터베이스 벤더를 지정한다.</p> <p>- Oracoe : oracle, Altibase: altibase</p>



	<ul style="list-style-type: none"> <li>● 설정예 DB_VENDER=oracle</li> </ul>
TP_SERVER_DOWN	<p>서버 프로세스가 실행되면서 ENPharos TP 에이전트에 접속하지 못한 경우에 서버 프로세스를 종료할 것인지 여부를 등록한다.</p> <ul style="list-style-type: none"> <li>● 설정예 true : 접속 오류 시, 서버 다운 false : 접속 오류 시, 서버 다운하지 않음</li> </ul>
LIBPATH	Pharos Link 에이전트 공유라이브러리 경로를 지정한다.
LD_LIBRARY_PATH	Pharos Link 에이전트 공유라이브러리 경로를 지정한다.

[표] 3-4 pharosadt.env 파일 내용

### 3.3.2. ENPharos TP 라이브러리 권한 변경

Tmax 관리자 계정과 ENPharos TP 에이전트 계정이 다른 경우에 ENPharos TP 에이전트에서 제공하는 라이브러리를 Tmax 서버 프로세스에서 참조하기 때문에 라이브러리에 대해서 실행 권한을 주어야 한다.

```
pharottp@AIX6:/home/pharostp/apm/lib> ls -al
-rwx----- 1 pharostp dba      232907 Apr 26 13:28 libpharos.so*
-rwx----- 1 pharostp dba      241867 Nov  7 15:58 libpharosdb.so*
-rwx----- 1 pharostp dba      120062 Apr 29 22:16 libpharostpcom.so*
-rwx----- 1 pharostp dba     111847 Apr 30 11:32 libpharostpdaehm.dbg.so*
-rwx----- 1 pharostp dba       73138 Apr 26 13:29 libpharostpdaehm.so*
-rwx----- 1 pharostp dba     1113368 Apr 27 02:19 libpharostpshm.dbg.so*
-rwx----- 1 pharostp dba      98878 Apr 26 13:29 libpharostpshm.so*
-rwx----- 1 pharostp dba      34045 Apr 27 02:19 libpharostpsh.dbg.so*
-rwx----- 1 pharostp dba       32099 Apr 26 13:29 libpharostpsh.so*
-rwx----- 1 pharostp dba      85985 Apr 26 13:30 libpsysrmon.dbg.so*
-rwx----- 1 pharostp dba      38597 Apr 26 13:30 libpsysrmon.so*
-rwx----- 1 pharostp dba     230604 Apr 26 13:30 libptpadmin.dbg.so*
-rwx----- 1 pharostp dba      84589 Apr 26 13:30 libptpadmin.so*
-rwx----- 1 pharostp dba       1540 Apr 26 13:28 libsigusrproc.so*

pharostp@AIX6:/home/pharostp/apm/lib> chmod 755 *.so
-rwxr-xr-x 1 pharostp dba      232907 Apr 26 13:28 libpharos.so*
-rwxr-xr-x 1 pharostp dba      241867 Nov  7 15:58 libpharosdb.so*
-rwxr-xr-x 1 pharostp dba      120062 Apr 29 22:16 libpharostpcom.so*
-rwxr-xr-x 1 pharostp dba     111847 Apr 30 11:32 libpharostpdaehm.dbg.so*
-rwxr-xr-x 1 pharostp dba       73138 Apr 26 13:29 libpharostpdaehm.so*
-rwxr-xr-x 1 pharostp dba     1113368 Apr 27 02:19 libpharostpshm.dbg.so*
-rwxr-xr-x 1 pharostp dba      98878 Apr 26 13:29 libpharostpshm.so*
-rwxr-xr-x 1 pharostp dba      34045 Apr 27 02:19 libpharostpsh.dbg.so*
-rwxr-xr-x 1 pharostp dba       32099 Apr 26 13:29 libpharostpsh.so*
-rwxr-xr-x 1 pharostp dba      85985 Apr 26 13:30 libpsysrmon.dbg.so*
-rwxr-xr-x 1 pharostp dba      38597 Apr 26 13:30 libpsysrmon.so*
-rwxr-xr-x 1 pharostp dba     230604 Apr 26 13:30 libptpadmin.dbg.so*
-rwxr-xr-x 1 pharostp dba      84589 Apr 26 13:30 libptpadmin.so*
-rwxr-xr-x 1 pharostp dba       1540 Apr 26 13:28 libsigusrproc.so*
```

### 3.3.3. ENPharos TP 관리자 계정의 환경설정 추가

ENPharos TP 관리자 계정의「.profile」에 ORACLE, Tmax 환경설정을 추가해 주어야 한다. ENPharos TP 에이전트에서 ORACLE 클라이언트 라이브러리와 Tuxedo 클라이언트 라이브러리를 참조하기 때문에 경로가 설정되어 있지 않거나, 다르면 ENPharos TP 에이전트 Boot 시, 오류가 발생한다.

#### 3.3.3.1. ORACLE 환경 설정

```
#=====#
# Oracle Environment
#=====#
export ORACLE_BASE=/oracle/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/10.2.0
export ORACLE_SID=KFCORAP1
export NLS_LANG=American_America.K016KSC5601
export NLS_DATE_FORMAT="YYYY/MM/DD"
export PATH=$PATH:$ORACLE_HOME/bin:$ORACLE_BASE/patch/OPatch
export LIBPATH=$LIBPATH:$ORACLE_HOME/lib:$ORACLE_HOME/lib:$LIBPATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

[그림] 3-24 ORACLE 환경 변수

#### 3.3.3.2. Tmax 환경 설정

```
#=====#
# Tmax Environment
#=====#
export TMAXDIR=/home/Tmax
export TMAX_HOST_ADDR=175.118.115.24
export TMAX_HOST_PORT=8888
export TMAX_CONNECT_TIMEOUT=3
export PATH=$PATH:/home/Tmax/Tmax5/bin
export LIBPATH=$LIBPATH:$TMAXDIR/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$TMAXDIR/lib
```

[그림] 3-25 Tmax 환경 변수

### 3.3.4. Tmax 관리자 계정의 환경변수 등록

Tmax 관리자 계정도 ENPharos TP 에이전트 환경을 적용해야 주어야 한다. 이는 ENPharos TP 라이브러리를 이용하여 Tuxedo의 정보를 수집하기 때문이다. 관리자 계정 「.profile」에 다음과 같이 환경변수를 추가 해준다.

```
# 기존 Tmax 계정의 환경 설정들
~
./home/pharostp/apm/bin/pharostp.env
```

[그림] 3-26 Tmax 계정 환경변수

「pharostp.env」 파일의 정보는 다음과 같다.

```
#=====#
# PHAROSTP TPM Environment ( TP-M environment set )
#=====#
export PHOMEDIR=/home/pharosTP/apm
export PHAROS_TPHOME=$PHOMEDIR/pharostp
export PHAROS_CONFIG=pharostp.cfg
export PHAROS_LOGDIR=$PHOMEDIR/log
export LOGLEVEL=EWIDT
export OS_VENDER=ibm64
export TP_TYPE=tmax
export DB_VENDER=oracle
export ORACLE_VERSION=oracle
export TP_SERVER_DOWN=false
export LIBPATH=$LIBPATH:$PHOMEDIR/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$PHOMEDIR/lib

#=====#
# export preload (tuxedo= libpharostpsh.so, tmax=libpharosload.so)
#=====#
if [ $OS_VENDER = ibm64 ]; then
    export LDR_PRELOAD64=$PHOMEDIR/lib/libpharosload.so
elif [ $OS_VENDER = ibm32 ]; then
    export LDR_PRELOAD=$PHOMEDIR/lib/libpharosload.so
elif [ $OS_VENDER = sun64 ]; then
    export LD_PRELOAD_64=$PHOMEDIR/lib/libpharosload.so
else
    export LD_PRELOAD=$PHOMEDIR/lib/libpharosload.so
fi
```

[그림] 3-27 pharostp.env 설정 내용

---

## 주의

Tmax 관리자 계정과 ENPharos TP 에이전트 계정이 다른 경우에 pharostp.env 파일을 Tmax 관리자 계정에서 참조하기 때문에 해당 파일에 대한 읽기 권한을 주어야 한다.

---

### 3.3.5. Tmax 시스템에 tmaxadminfosvc 서버 등록

1. Tmax 시스템인 경우 ENPharos TP에서 tmaxadmin 정보를 조회하기 위해 서버와 서비스를 추가해야 한다. Tmax 환경파일에 다음과 같이 추가 해준다.

```
● 서버 절에 추가
*SERVER
## Pharos Server ##
tmaxadminfosvc      SVGNAME = svg1, MIN=1, MAX=1,
                     CLOPT="-o $(SVR).$(CDATE).log -e $(SVR).$(CDATE).log"

● 서비스 절에 추가
*SERVICE
## Pharos Service ##
TMAXADMINFOSVC      SVRNAME = tmaxadminfosvc
```

[그림] 3-28 Tmax 시스템의 서비스 등록 방법

2. 서버 절과 서비스 절을 추가 한 다음 Tmax 환경파일을 컴파일 한다.

```
Tmax@AIX6:/home/Tmax/config> cfl -i xxx.m
CFL is done successfully for node(faust)
Tmax@AIX6:/home/Tmax/config> gst
SVC tables are successfully generated
GST is successfully done
```

3. Tmax의 서버프로그램이 위치한 곳에 ENPharos TP 에이전트에 있는 tmaxadminfosvc 파일을 복사한 후 Tmax 시스템을 재기동 한다.

```
Tmax@AIX6:/home/Tmax/appbin> cp /home/pharostp/apm/bin/tmaxadminfosvc .
Tmax@AIX6:/home/Tmax/appbin> ls tmaxadminfosvc
tmaxadminfosvc
```

### 3.3.6. ENPharos TP 관리자 계정의 tmax.env 설정

ENPharos TP 관리자 계정에서 「tmax.env」 파일을 설정 해 주어야 하는데, 이는 Tmax에 등록된 서비스를 호출하기 위해서이다. Tmax 관리자 계정에 있는 「tmax.env」와 동일하게 설정해 주거나 동일 파일을 복사해 온다.

```
pharostp@AIX6:/home/pharostp/apm/bin> vi tmax.env
```

```
[TMAX]
TMAXDIR=/home/Tmax
TMAX_HOST_ADDR=175.118.XXX.XXX
TMAX_HOST_PORT=8888
SDLFILE=/home/Tmax/sample/sdl/tmax.sdl
FDLFILE=/home /Tmax/sample/fdl/tmax.fdl
TMAX_CONNECT_TIMEOUT=2
```

[그림] 3-29 ENPharos TP tmax.env 설정

### 3.3.7. ENPharos TP 설정파일 수정(pharostp.cfg)

```
pharostp@AIX6:/home/pharostp/apm/config> ls
customgen.sample.cfg  pharostp.sample.cfg
pharostp@AIX6:/home/pharostp/apm/config> cp pharostp.sample.cfg pharostp.cfg
pharostp@AIX6:/home/pharostp/apm/config> ls
customgen.sample.cfg  pharostp.cfg          pharostp.sample.cfg
```

Pharostp.cfg 설정 방법은 Tuxedo 와 동일 하므로, “3.2.5. ENPharos TP 설정파일 수정 (pharostp.cfg)”를 참조한다.

※ ENPharos TP와 Tmax가 동일한 관리자 계정에 설치 되어 있다면 pharostp.env 파일만 환경변수에 등록해 주도록 한다.

### 3.3.8. psysrmon 생성

psysrmon은 시스템에 대한 CPU와 메모리 정보를 조회하는 프로세스이다. 그러나 시스템에 대한 CPU와 메모리 정보를 시스템 벤더에 따라서 다양한 API를 제공한다. 또한 동일한 API라도 API에서 제공되는 구조체가 다양하기 때문에 해당 시스템에 맞게 compile 되어야 한다.

ENPharos TP 에이전트에서 제공하는 라이브러리 「psysrmon.so」와 「psysrmon.c」 소스를 컴파일하여 링크 하면 「psysrmon」 실행 파일이 생성한다.

## 3.4. 에이전트 기동과 종료

### 3.4.1. 에이전트 기동

환경변수 설정이 완료되면 ENPharos TP 에이전트를 실행할 수 있다. ptpboot 시, 오류가 난다면 “ENPharos TP Administration Guide”의 장애조치 부분을 참조한다.

```
$ ptpboot
```

```
Pharos Manager process starting ...  
Pharos Adapter process starting ...  
ENPharos TP Admin process starting ...  
Pharos System Monitoring process starting ...
```

### 3.4.2. 에이전트 종료

ENPharos TP 에이전트 종료는 간단하게 ptpdown 명령만 실행하면 된다. ENPharos TP 에이전트가 종료되면 자동적으로 TP 서버에 적용된 거래 추적 시스템은 종료된다. 즉, 거래 추적 레벨인 "trace\_level"이 0으로 자동 변경된다.

```
$ ptpdown
```

```
*WARNING* [pharostp] may be killed: pid=[987192]  
*WARNING* [ptpadmin] may be killed: pid=[520260]  
*WARNING* [psysrmon] may be killed: pid=[716834]  
*WARNING* [pmm] may be killed: pid=[467058]
```

## 3.5. UI 접속

수집서버가 정상적으로 실행되고 에이전트를 적용시킨 대상 TP가 기동 되었다면, UI에 접속하여 모니터링 화면으로 에이전트 설치가 제대로 되었는지 확인하여 본다. 웹 브라우저를 열어 <http://{PharosServerIP}:45000> 에 접속하면 아래와 같이 ENPharos TP UI를 확인 할 수 있다. User ID/Password 에 admin/admin을 입력하고 로그인 하여 사용자 화면이 정상적으로 보여지면 정상적으로 설치가 된 것이다.



[그림] 3-30 ENPharos TP UI 로그인 화면

## 참고

ENPharos TP UI를 실행하기 위해서는 Adobe Flash Player(11.9 버전 이상)가 클라이언트 PC에 사전 설치 되어 있어야 한다.

포트번호는 pharos\_env 파일의 PORT\_TOMCAT\_HTTP에서 정의한 포트번호를 입력해야 한다.

## 3.6. 에이전트 제거

다음은 ENPharos TP 에이전트 제거 과정이다.

1. ENPharos TP 관리자 계정으로 로그인하여 ENPharos TP 에이전트를 종료한다.

```
$ ptpdown

*WARNING* [pharostp] may be killed: pid=[987192]
*WARNING* [ptpadmin] may be killed: pid=[520260]
*WARNING* [psysrmon] may be killed: pid=[716834]
*WARNING* [pmm] may be killed: pid=[467058]
```

2. TP Monitor 시스템을 종료한다. (Tuxedo, Tmax)

```
$ tmdown (tuxedo: tmsshutdown).
```

3. Tmax Config 정보를 삭제한다.

```
## PHAROS SERVER
tmaxadminfosvc SVGNAME = svg1, MIN=1, MAX=1, RESTART=Y,
                CLOPT = "-o $(CYEAR)$(CMONTH)$(CDAY)/$(SVR).$(DATE).log
                        -e $(CYEAR)$(CMONTH)$(CDAY)/$(SVR).$(DATE).log"

## PHAROS SERVICE
TMAXADMINFOSVC SVRNAME = tmaxadminfosvc
```

[그림] 3-31 Tmax conifg – 서비스 제거

4. TP Monitor 관리자 계정의 「.profile」에서 등록한 환경변수를 제거한다.

```
#. /home/pharosTP/apm/bin/pharostp.env
```

[그림] 3-32 pharostp.env 제거

5. ENPharos TP 관리자 계정의 「.profile」에서 등록한 환경변수를 제거한다.

```
#=====#
# pharosTP Environment
#=====#
#. /home/pharostp/bin/pharosadt.env
```

[그림] 3-33 pharosadt.env

6. ENPharos TP 계정으로 생성된 큐와 공유 메모리를 삭제한다.

```
$ ipcs
메시지 대기행렬 (Message Queues) :
  Id      key      mode      owner      group
q 2097185 0x00009c24 -Rrw-rw-rw- pharostp  staff

공유 메모리 (Shared Memory Segments) :
  Id      key      mode      owner      group
m 1048578 0x00015f87 -Rrw-rw-rw- pharostp  staff

$ipcrm -q 2097185
$ipcrm -m 1048578
```

7. ENPharos TP 계정의 디렉토리를 제거한다.

```
$ rm -rf /home/pharostp
```

8. ENPharos TP 관리자 계정을 삭제한다.

```
$ userdel pharostp
```