

# *ES/1 NEO*

CSシリーズ

Acquire 使用者の手引き

～Unix/Linux 編～

12.50



株式会社 アイ・アイ・エム

第1版 2024年2月

©著作権所有者 株式会社 アイ・アイ・エム 2024年

© COPYRIGHT IIM CORPORATION, 2024

ALL RIGHT RESERVED. NO PART OF THIS PUBLICATION MAY  
REPRODUCED OR TRANSMITTED IN ANY FORM BY ANY MEANS,  
ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPY RECORDING,  
OR ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM WITHOUT  
PERMISSION IN WRITING FROM THE PUBLISHER.

“RESTRICTED MATERIAL OF IIM “LICENSED MATERIALS – PROPERTY OF IIM

# 目次

第 1 章 事前準備 .....	1
1.1. 導入の前提条件 .....	1
1.1.1. OS1 .....	1
1.1.2. インストール先のディスク .....	1
1.1.3. ディスクスペース .....	1
1.1.4. コマンド .....	2
1.1.5. 通信 .....	3
1.1.6. その他 .....	3
1.1.7. Oracle .....	3
1.1.8. Symfoware .....	3
1.2. システム管理者による事前準備 .....	4
1.2.1. Acquire 用ユーザ ID の作成 .....	4
1.2.2. cron の許可登録 .....	4
1.3. システム管理者による課金機能の事前準備 .....	5
1.3.1. 課金機能の事前準備 .....	5
1.3.2. 課金機能の自動起動 .....	8
1.3.3. pacct ファイル自動削除運用のご提案 .....	11
1.4. システム管理者による Oracle データ収集の事前準備 .....	14
1.4.1. 前提条件 .....	14
1.4.2. 事前準備 .....	14
第 2 章 インストール .....	16
2.1. インストール形態 .....	16
2.1.1. Acquire のインストール形態 .....	16
2.2. インストール手順 .....	17
2.2.1. サイレント形式のインストール .....	17
2.3. Oracle データの収集設定 .....	18
2.3.1. Oracle 関連の環境変数のセット .....	18
2.3.2. Oracle Wallet 配置ディレクトリの作成 .....	18
2.3.3. Oracle Wallet の作成 .....	18
2.3.4. Acquire 用ネットサービス名の設定 .....	19
2.3.5. Acquire 用 sqlnet 環境の設定 .....	20
2.3.6. Acquire 用 Oracle ユーザ作成 SQL ファイルの編集 .....	21
2.3.7. Acquire 用 Oracle ユーザ作成 SQL ファイルの実行 .....	26
2.3.8. Oracle Wallet への認証情報の登録 .....	28
2.3.9. Oracle への接続確認 .....	28
2.4. Symfoware データの収集設定 .....	30
2.4.1. Symfoware データを収集する場合の設定 .....	30
第 3 章 アンインストール .....	32
3.1. Acquire のアンインストール手順 .....	32
3.1.1. 削除手順 .....	32

## 第1章 事前準備

ここでは、ES/1 NEO CS シリーズの Unix/Linux データ収集モジュールである Acquire 導入前の確認項目と作業内容を説明します。事前準備の中で「root」ユーザで作業する部分があります。

### 1.1. 導入の前提条件

#### **注意！**

仮想化環境において、既に Acquire をインストールしてデータ収集を開始しているゲスト OS のクローンを作成しないでください。データ収集を開始しているゲスト OS のクローンを Control Center の処理対象とした場合、Control Center が誤作動を起こす可能性があります。

#### 1.1.1. OS

パフォーマンス管理対象の OS については、「サポート環境」の「Unix サーバ」「Linux サーバ」をご参照ください。

#### 1.1.2. インストール先のディスク

Acquire を対象サーバのローカルディスクに導入可能であること。

#### **注意！**

クラスタ環境等の複数サーバで共用しているディスクや、ネットワークストレージには Acquire を導入できません。

#### 1.1.3. ディスクスペース

Acquire 実行プログラム用および収集データを格納するディスクスペースとして 1GB を確保してください。収集間隔や取得対象とするデータの種類により、必要となるディスクスペースは増減します。

なお、収集データのサイズは対象システムの環境によって変化します。

#### **注意！**

この容量は標準的な構成および設定を行った場合に推奨される値で、取得する性能情報に応じて増減します。収集間隔またはデータ保存期間を変更する場合のディスク容量の見積もりは弊社 SE にご連絡ください。

### 1.1.4. コマンド

各 OS に対して○のついているコマンドが使用できる状態であること。

accton はユーザ・コマンド毎のプロセッサ使用率を収集する場合のみ必要。

df はファイルシステムのスペース使用率を取得する場合のみ必要。

	sar	vmstat	iostat	mpstat	netstat	entstat	lparstat	ps	accton	df	gzip	compress	bc
AIX 7.1-7.3	○	○	○	×	○	○	○	○	○	○	○	×	×
HP-UX 11i v3	○	○	×	×	○	×	×	○	○	○	○	×	×
Solaris 11.0-11.4	○	○	○	○	○	×	×	○	○	○	○	×	×
RHEL 7.0-7.9, RHEL 8.0-8.8, RHEL 9.0-9.2 Oracle Linux 7.0-7.2 Oracle Linux 7.7/7.9	○	○	○	×	×	×	×	○	○	○	○	×	○
SLES 12.0-12.5, SLES 15.0-15.4	○	○	○	×	×	×	×	○	○	○	○	×	○

#### 注意！

Linux では、データ収集に必要なコマンドがインストールされていない場合があります。括弧内はそれぞれのパッケージに含まれるコマンドです。

sysstat (sar、iostat) 、bc (bc) : すべての Linux で必要なパッケージです。

psacct または acct (accton) : ユーザ・コマンド毎のプロセッサ使用率を収集する場合のみ必要なパッケージです。

まず rpm コマンドで各パッケージの有無を確認してください。

例) rpm -q sysstat

パッケージがインストールされていない場合、必ずディストリビューション標準のパッケージをインストールしてください。ディストリビューション標準のパッケージは、Linux インストールメディアか、ディストリビュータの Web ページから入手可能です。

なお、パッケージはそれぞれのディストリビューションのバージョンで標準提供されるものを使用して下さい。

例) rpm -ivh sysstat-4.0.2-1.i386.rpm

#### 注意！

AIX 環境で収集されたデータにおいて、sar コマンドと iostat コマンドで報告されるデバイス名が一致していない場合があります。このようなデータは Control Center がデバイスデータの変換処理を行いません。そのため、該当環境ではフラットファイルの「デバイス(Unix) : ATDEV」データがすべて欠損します。

例)

iostat コマンドの出力

asm/dev000001-

asm/dev000002-

asm/dev000003-

sar コマンドの出力

asm/dev000001-93

asm/dev000002-93

asm/dev000003-93

### 1.1.5. 通信

---

管理用マシンからデータ収集対象サーバに対して、FTP または SFTP を利用して接続が行えること。FTP を利用する場合には、バイナリモードとアスキーモードの両方が許可されていること。日本語のメッセージを返す FTP または SFTP サーバに関しては、EUC 文字コードが使用できること。

#### **注意！**

アスキーモードがデフォルトで許可されていない FTP サーバもありますので、設定ファイルの内容を確認してください。

（特に vsftpd はアスキーモードが不許可になっている事例が多数あります。）

Solaris において、日本語のメッセージを返し、言語設定の初期値が UTF-8 である FTP サーバが確認されています。その場合は、言語設定を EUC に変更してください。

### 1.1.6. その他

---

cron から送られるメールが正しく送信されること。

正常に Acquire が動作しない為、Acquire が使用する各種のディレクトリにはリンクの設定を行わないこと。

### 1.1.7. Oracle

---

Oracle バージョンについては、「サポート環境」の「Oracle」をご参照ください。

### 1.1.8. Symfoware

---

Symfoware バージョンについては、「サポート環境」の「Symfoware」をご参照ください。

#### **注意！**

Acquire は、Symfoware Server のセキュリティ運用形態が「標準セキュリティ運用」の場合、データを収集できません。「標準運用」の場合のみ、データ収集可能です。

## 1.2. システム管理者による事前準備

Acquire のインストールにあたって必須となる事前作業を説明いたします。

### 1.2.1. Acquire 用ユーザ ID の作成

Acquire に対する専用ユーザ ID を作成します。このユーザ ID でログインして作業を進めます。

ユーザ名	: athene (任意の名前で結構です。本資料では以下 athene として説明します)		
パスワード	: 半角 15 文字までの任意のパスワードを必ず設定してください。但し、パスワードに「,」(カンマ) は含まないでください (パスワード無しは不可)。		
プライマリグループ	: Linux の場合	→	users
	AIX、HP-UX、Solaris の場合	→	adm
セカンダリグループ	: oinstall (セカンダリグループは Oracle データを取得する場合のみ設定。Oracle データを取得しない場合は設定不要)		
デフォルトシェル	: Linux の場合	→	bash (csh は不可)
	AIX、HP-UX、Solaris の場合	→	sh または ksh (csh は不可)

### 1.2.2. cron の許可登録

Acquire は cron の機能を使用します。athene で cron 機能を利用できるようにする為に、cron.allow か cron.deny ファイルの設定を行います。これらのファイルは/etc/cron.d/または/etcの下にあります。

(1)cron.allow ファイルが存在する場合

cron.allow に記述されているユーザ ID のみ cron 機能を使用できます。cron.allow に athene を追加します。

(2)cron.deny ファイルが存在する場合

cron.deny に記述されている以外のユーザ ID が cron 機能を使用できます。この場合は何もする必要はありません。

### 1.3. システム管理者による課金機能の事前準備

ユーザ・コマンド毎のプロセッサ使用率を収集する場合、システム管理者は課金機能の事前作業を行う必要があります。ユーザ・コマンド毎のプロセッサ使用率を収集しない場合は不要な作業となります。

#### 1.3.1. 課金機能の事前準備

##### ①Metron.save.d ディレクトリの作成

Metron.save.d というディレクトリを、pacct ファイルと同階層のディレクトリに作成します。pacct ファイルがあるディレクトリは、OS 毎に異なります。

AIX の場合	→/var/adm/Metron.save.d /var/adm に pacct ファイルが無い場合は、下記コマンドで事前に pacct ファイルを作成しておいてください。 touch /var/adm/pacct
---------	--

HP-UX、Solaris の場合	→/var/adm/Metron.save.d
Linux の場合	→/var/account/Metron.save.d または /var/adm/Metron.save.d pacct ファイルの場所は rpm コマンドから確認できます。 rpm -ql psacct または rpm -ql acct

##### ②Metron.save.d ディレクトリのパーミッション

オーナー	: athene	
グループ ID	: AIX、HP-UX、Solaris の場合	→adm
	Linux の場合	→users
パーミッション	: 755	

##### ③pacct ファイルのパーミッション

課金情報を Acquire が収集する為に、pacct ファイルのアクセス権限を変更してください。

AIX、Solaris、HP-UX の場合	→パーミッション : 644
RHEL7/8/9、OracleLinux7、SLES12/15 の場合	→パーミッション : 664、グループ ID : users
その他の Linux の場合	→パーミッション : 644



#### ④課金機能の起動、停止

Acquire は Linux の課金情報からユーザ／コマンドデータを取得しますので、以下のコマンドを root から実行して課金機能を起動します。

##### ●課金機能の起動

AIX、HP-UX、Solaris の場合 →/usr/lib/acct/accton /var/adm/pacct

RHEL7/8/9、OracleLinux7、SLES12/15 以降の場合

#systemctl list-unit-files --type=service | grep acct

結果が「psacct.service disabled」場合には、次のコマンドにて収集サービスを起動します。

systemctl start psacct.service

その他の Linux の場合 →/usr/sbin/accton /var/account/pacct または  
/usr/sbin/accton /var/adm/pacct

課金機能が動作していることを確認する為に、ls -l コマンドにて pacct ファイルのサイズが更新されることを確認してください。

課金機能を停止するには、ファイル名を指定せずに accton コマンドを実行します。パフォーマンスデータを取得している間、課金機能を停止することはありません。Acquire のアンインストール時などに実行します。

##### ●課金機能の停止

AIX、HP-UX、Solaris の場合 →/usr/lib/acct/accton

RHEL7/8/9、OracleLinux7、SLES12/15 の場合 →/usr/sbin/accton off

その他の Linux の場合 →/usr/sbin/accton

**注意！**

- 1) 上記は一般的な課金機能起動方法になります。上記の方法で課金機能が起動しない場合があります。  
詳細につきましては必ずメーカー提供のマニュアルにてご確認ください。
- 2) 課金情報は、システム環境によってデータ量が膨大になることがあります。/var ファイルシステムに十分な容量があることを確認してから課金機能を稼働させてください。/var ファイルシステムがいっぱいになると、課金機能が停止してそれ以降のデータが取得できなくなります。
- 3) OS によっては pacct ファイルのサイズが上限に達すると、課金機能が停止してそれ以降のデータが取得できなくなります。例えば、HP-UX では pacct ファイルの上限サイズが約 2.5MB と比較的小さい値になっているため、サーバの稼働状況によっては上限に達し易くなる傾向があります。なお、pacct ファイルの上限サイズなどの条件は OS のバージョンによって変更される可能性があります。課金機能をご使用される際には、必ずメーカー提供のマニュアルをご確認いただき、運用についてあらかじめご検討ください。
- 4) プロセス名に[ ]が付加されているプロセスについては、ES/1 NEO CS シリーズの仕様として、プロセス名から[ ]を削除して報告します。  
例. [kswapd] → kswapd
- 5) ③の Linux の pacct ファイルのパーミッションについて、ディストリビューションによって設定値が異なるのは、protected\_symlinks 及び protected\_hardlinks の設定の有無に起因しています。  
Acquire は pacct ファイルのハードリンクを作成します。protected\_symlinks 及び protected\_hardlinks が設定されている場合は、pacct ファイルに対する Acquire ユーザのアクセス権限を強める必要があります。  
RHEL7/8/9、SLES12/15 などは protected\_symlinks 及び protected\_hardlinks がデフォルトで設定されています。環境によってはこの他のディストリビューションでも設定されている場合があります。ご注意ください。

**設定されている場合**

```
# cat /proc/sys/fs/protected_symlinks
1
# cat /proc/sys/fs/protected_hardlinks
1
```

グループ ID : users

パーミッション : 664

**設定されていない場合**

```
# cat /proc/sys/fs/protected_symlinks
0
# cat /proc/sys/fs/protected_hardlinks
0
```

グループ ID : 変更不要

パーミッション : 644

### 1.3.2. 課金機能の自動起動

システムをリブート後、自動的に課金機能を開始させるには、OS 側での設定が必要です。課金機能の自動起動の設定方法は、OS によって異なります。ここでは、代表的な OS の例を紹介します。

なお、ここでご紹介する設定に関しまして、弊社ではいかなる責任も負いかねます。課金機能の自動起動に関しては、各 OS のマニュアルなどをご確認の上、お客様の環境に即したものを設定してください。

#### [AIX7.1～7.3]

/etc/rc ファイルを編集します。/etc/rc ファイルには書き込み権限がありませんので、一時的にパーミッションを変更するか、編集後に強制保存を行ってください。

/etc/rc に下記の内容を追記し、保存してください。

```
/usr/bin/su - adm -c /usr/sbin/acct/startup
```

#### [HP-UX11i]

root ユーザから、/etc/rc.config.d/acct を編集し、課金機能の自動起動を設定します。

/etc/rc.config.d/acct ファイルには書き込み権限がありませんので、一時的にパーミッションを変更するか、編集後に強制保存を行ってください。

/etc/rc.config.d/acct 中の START\_ACCT=を 0 から 1 に変更してください。

```
START_ACCT=1
```

#### [Solaris11]

root ユーザから作業を行ってください。

/etc/init.d に任意の名前の起動スクリプトを作成します。ここでは、pacct という名前のファイルを作成します。

```
touch /etc/init.d/pacct  
chown root:sys /etc/init.d/pacct  
chmod 744 /etc/init.d/pacct
```

次に起動スクリプトを編集します。/etc/init.d/pacct に下記の内容を記述し、保存してください。

```
#!/sbin/sh
state="$1"
case "$state" in
'start' )
    echo 'Starting process accounting'
    /usr/lib/acct/accton /var/adm/pacct
    ;;
'stop')
    echo 'Stopping process accounting'
    /usr/lib/acct/accton
    ;;
*)
    echo "Usage: $0 { start | stop }"
    exit 1
    ;;
esac
exit 0
```

作成した起動スクリプトが、システム起動時と終了時にそれぞれ実行されるように設定します。

```
cp /etc/init.d/pacct /etc/rc2.d/S99pacct
chown root:sys /etc/rc2.d/S99pacct
chmod 744 /etc/rc2.d/S99pacct
cp /etc/init.d/pacct /etc/rc0.d/K10pacct
chown root:sys /etc/rc0.d/K10pacct
chmod 744 /etc/rc0.d/K10pacct
```

[RHEL7/8/9、OracleLinux7、SLES12/15]

root ユーザから psacct.service の状態を確認してください。

```
# systemctl list-unit-files --type=service | grep acct
```

下記のように enabled になっていれば有効になっています。

作業はここで終了です。

```
psacct.service (SLES の場合は acct.service)          enabled
```

下記のように disabled になっているか、該当のファイルがない場合は、有効化する必要があります。

```
psacct.service (SLES の場合は acct.service)          disabled
```

自動起動は下記コマンドから設定してください。

```
# systemctl enable psacct.service (SLES の場合は acct.service)
```

## [その他の Linux]

root ユーザから起動スクリプトが起動時に有効になっているか確認します。

下記のようにランレベル 3 が on になっていれば有効になっています。

作業はここで終了です。

```
# chkconfig --list | grep psacct (SLES の場合は acct)
```

```
psacct      0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

下記のようにランレベル 3 が off になっていれば、有効化する必要があります。

```
# chkconfig --list | grep psacct (SLES の場合は acct)
```

```
psacct      0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

課金機能の起動スクリプトを有効にするには下記コマンドを発行してください。

```
# chkconfig psacct on (SLES の場合は acct)
```

### 1.3.3. pacct ファイル自動削除運用のご提案

pacct ファイルの自動削除運用をご提案します。pacct ファイルが肥大化すると、課金情報の収集が停止してしまいます。自動削除の運用方法は Linux とそれ以外の OS で異なります。

[AIX、HP-UX、Solaris]

以下の手順を実行してください。

- ①root でログインします。
- ②/usr/lib/acct ディレクトリの下に、次のような内容のスクリプトファイル「rmpacct」を作成します。

rm -f /var/adm/pacct?	過去の pacct ファイル群を削除
rm -f /var/adm/pacct??	過去の pacct ファイル群を削除

- ③下記のようにコマンド（ckpacct）およびスクリプト（rmpacct）を root の crontab に設定します。

```
59 * * * * /usr/lib/acct/ckpacct
1 * * * * sh /usr/lib/acct/rmpacct
```

ckpacct コマンドは、/var/adm/pacct の大きさをチェックして一定の大きさを超えている場合に、pacct ファイルの連番コピー（pacct1,pacct2,・・・）を作成して、/var/adm/pacct のサイズをリセットします。また、/var ファイルシステムの空き容量が一定サイズ以下になった場合、課金機能を停止します。

この運用例では、毎時 59 分に ckpacct コマンドにより pacct ファイルの大きさをチェックします。また、毎時 1 分に rmpacct スクリプトが起動して、過去の pacct ファイルを削除します。これらにより、/var ファイルシステム上で、課金情報のスペースを抑えることができます。

なお、HP-UX のように pacct ファイルのサイズが上限に達し易いサーバの場合は、課金機能の運用が途中停止するのを避けるために、下記の通り ckpacct の実行頻度を増やすことをご検討ください。

<30 分に 1 回実行する例>

```
29,59 * * * * /usr/lib/acct/ckpacct
1,31 * * * * sh /usr/lib/acct/rmpacct
```

[Linux]

logrotate というログのローテーション実行ツールを使用して運用します。

- ①/etc/logrotate.d ディレクトリの下にある psacct ファイル（SLES の場合は acct）を編集してください。多くのディストリビューションでは、psacct ファイルは psacct パッケージと同時にインストールされます。ファイルがすでに存在するか「rpm-ql psacct」などで確認してください。

ファイル名 : psacct （SLES の場合は acct）  
 オーナー : root  
 グループ ID : root  
 パーミッション : 644

- ②/etc/logrotate.d/psacct ファイルを以下のように編集します。

```

～前略～
compress
notifempty
  daily          ←1 日 1 回実行します。
  rotate 2       ←この例では 2 世代残します。最低 1 世代必要です。
  create 0644 root root ←psacct ファイルのパーミッションを 644 とします。

～後略～
endscript
}
```

※RHEL7/8/9、OracleLinux7、SLES12/15 の場合、create から始まる記述を下記のように変更してください。  
 create 0664 root users ←psacct ファイルのパーミッションを 664 とし、グループ を users にします。

この設定では、1 日 1 回 psacct ファイルを自動でバックアップして削除します。「create 0600 root root」となっていると Acquire が課金情報の収集を行えなくなりますのでご注意ください。  
 また、logrotate 自体が毎日実行されるように設定されているかご確認ください。詳細は logrotate のマニュアルをご覧ください。

logrotate は、cron.daily に登録されています。24 時間以内に psacct ファイルが大きくなりすぎてしまう場合は、logrotate を下記のように root の crontab に登録し、24 時間より短い間隔で実行します。

SLES の場合

```
3 * * * * /usr/sbin/logrotate -f /etc/logrotate.d/acct
```

RHEL、Oracle Linux の場合

```
3 * * * * /usr/sbin/logrotate -f /etc/logrotate.d/psacct
```

この運用を設定する場合、cron.daily による pacct のローテーションを無効にしてください。/etc/logrotate.conf を下記のとおり編集し無効にします。

**編集前**

```
# RPM packages drop log rotation information into this directory
include /etc/logrotate.d
```

**編集後****SLES の場合**

```
# RPM packages drop log rotation information into this directory
tabooext +acct
include /etc/logrotate.d
```

**RHEL、Oracle Linux の場合**

```
# RPM packages drop log rotation information into this directory
tabooext +psacct
include /etc/logrotate.d
```

tabooext は include の前に設定する必要があります。include の後方で設定すると無効になりません。

**注意！**

これらの運用案は動作を保証するものではありません。システムに即した運用を検討してください。



## 1.4. システム管理者による Oracle データ収集の事前準備

ここでは、Oracle データを取得する場合の前提条件、およびシステム管理者が行わなければならない事前準備について説明します。

### 1.4.1. 前提条件

Acquire は、Oracle Wallet を使用してリスナー経由で対象 Oracle データベースに接続します。

そのため、以下の前提条件を満たしていることが必要です。

- Oracle Wallet が使用可能なこと。
- Oracle リスナーが設定済みで稼働していること。
- TCP/IP プロトコルがサポートされていること。

#### <参考 : Oracle Wallet セキュアな外部パスワード・ストア>

アプリケーションでデータベースへの接続にパスワード資格証明が使用される大規模なデプロイメントでは、コマンドラインを使用するかわりに、そのような資格証明をクライアント側の Oracle Wallet に格納できます。Oracle Wallet は、認証および署名資格証明の格納に使用されるセキュアなソフトウェア・コンテナです。

データベース・パスワード資格証明をクライアント側の Oracle Wallet に格納すると、ユーザー名およびパスワードをアプリケーション・コード、バッチ・ジョブまたはスクリプトに埋め込む必要がなくなります。これによって、パスワードをスクリプトおよびアプリケーション・コードに平文で公開する危険が少なくなり、ユーザー名およびパスワードの変更のたびにコードを変更する必要がないため、メンテナンスが簡単になります。また、アプリケーション・コードを変更する必要がないため、これらのユーザー・アカウントに対するパスワード管理方針も簡単に適用できます。

### 1.4.2. 事前準備

#### 1. ドメイン名／SID／データベース名／Oracle ホームのパス

Oracle のデータを収集する場合は、以下の情報を事前に確認してください。

##### ①ドメイン名

<非 CDB 構成/CDB 構成>

未設定またはデフォルトの場合は「WORLD」になります。

##### ②Oracle\_SID

<非 CDB 構成/CDB 構成>

SID は init(SID).ora ファイルまたは spfile(SID).ora にて確認できます。

##### ③データベース名

<非 CDB 構成>

データベース名は SQL\*Plus から以下の SELECT 文を実行して確認します。

```
select name from v$database;
```

<CDB 構成-CDB>

CDB 名は SQL\*Plus から以下の SELECT 文を実行して確認します。

```
select name from v$database;
```

<CDB 構成-PDB>

PDB 名は、SQL\*Plus にて CDB にログインした状態で以下の SELECT 文を実行し、表示される一覧から対象の PDB を確認します。

```
select name, open_mode from v$pdbs;
```

#### ④Oracle ホームのパス

##### <非 CDB 構成/CDB 構成>

環境変数「ORACLE\_HOME」に設定されている Oracle ホームのパスを確認します。

(例) /u01/app/oracle/product/12.2.0/dbhome\_1

#### 2. SQL\*plus の実行可否確認

Acquire は SQL\*Plus を起動して Oracle に接続します。

そのため、Acquire 用ユーザ ID で対象 Unix/Linux サーバにログインして、以下のコマンドが実行できることが必要です。

```
export ORACLE_HOME=<Oracle ホームディレクトリ>
export ORACLE_SID=<Oracle SID>
$ORACLE_HOME/bin/sqlplus /nolog
```

#### 3. Oracle 起動パラメータファイルの設定

Oracle の CPU 使用時間等の時間に関する統計データを収集できるようにする為、Oracle の起動パラメータファイルに TIMED\_STATISTICS パラメータを設定します。

なお、TIMED\_STATISTICS パラメータを設定しなくても、時間以外の統計データは収集することができます。

##### <非 CDB 構成/CDB 構成-CDB>

Oracle の起動パラメータファイルとして、spfile.ora か init.ora のどちらを使用しているかによって設定内容が変わります。

##### ・spfile(SID).ora を使用している場合

SQL\*Plus から以下の ALTER コマンドを発行することにより、動的にパラメータが有効になります。また、spfile.ora にも設定が反映されますので、Oracle の再起動後も設定が継続されます。

```
ALTER SYSTEM SET TIMED_STATISTICS=TRUE SCOPE=BOTH;
```

##### ・init(SID).ora を使用している場合

SQL\*Plus から以下の ALTER コマンドを発行することにより、再起動することなく動的にパラメータ変更ができます。

```
ALTER SYSTEM SET TIMED_STATISTICS=TRUE;
```

##### **注意！**

**ALTER コマンドによる設定では init.ora ファイルに反映されない為、一時的な設定変更になります。継続的に統計データを収集する場合は、init.ora ファイルを変更する必要があります。**

##### <CDB 構成-PDB>

SQL\*Plus から以下の ALTER コマンドを発行することにより、再起動することなく動的にパラメータ変更ができます。

```
ALTER SYSTEM SET TIMED_STATISTICS=TRUE SCOPE=BOTH;
```

#### 4. Acquire 用 Oracle ユーザのパスワード有効期限

Acquire は、パスワード認証タイプの Oracle ユーザを使用して Oracle にログインします。Oracle12c 以降では、パスワード認証タイプのユーザにおけるデフォルトのパスワード有効期限が「180 日」です。継続してデータ収集を行うために、Acquire 用 Oracle ユーザのパスワード有効期限が「UNLIMITED」になっている必要があります。

## 第2章 インストール

ここでは Acquire のインストール及び設定手順を説明します。Acquire のインストール処理では途中でシステム情報の入力を要求される部分がありますので、事前に記入していただいた事前作業シートの各項目を参照して作業を進めてください。

### 2.1. インストール形態

#### 2.1.1. Acquire のインストール形態

Acquire の提供媒体は DVD になります。DVD を直接 Unix/Linux サーバにマウントするか、PC からファイル転送して、インストール用ファイルを Unix/Linux サーバの Acquire 専用ユーザのホームディレクトリ直下にコピーします。なお、ファイル転送を行う場合は、必ずバイナリーモードで行ってください。インストール用ファイルの名前は、OS と CPU の組合せによって異なります。

ファイルコピー後、インストール用ファイルのオーナーを確認します。

オーナー : athene

もし、上記のオーナーになっていない場合は変更してください。

次に、telnet などの端末を利用してインストール作業を行います。

各項目の組み合わせについては下記の表をご参照ください。DVD の「x:¥ 12.50AC01¥モジュール格納フォルダ」にインストールモジュールが入っています。

OS	OS バージョン	CPU 種別	形式	モジュール名	モジュール 格納フォルダ	Acquire バージョン	インストール 手順
AIX	7.1-7.3	POWER	サイレント	acquireaix.tar	AIX125_VN70	12.50	2.2.1
HP-UX	11i v3	IPF	サイレント	acquirehpux.tar	HP125_VN71	12.50	2.2.1
Solaris	11.0-11.4	SPARC	サイレント	acquiresol8sparc.tar	Sol125_VN66_1	12.50	2.2.1
		x86、x64		acquiresol10contintel.tar	Sol125_VN66_2		
RHEL	7.0-7.9, 8.0-8.8, 9.0-9.2	x64	サイレント	acquirelin26x64.tar	Lin125_VN69_1	12.50	2.2.1
		z		acquirezlin26rhel7.tar	Lin125_VN69_3		
Oracle Linux	7.0-7.2, 7.7 / 7.9	x64	サイレント	acquirelin26x64.tar	Lin125_VN69_1	12.50	2.2.1
SLES	12.0-12.5, 15.0-15.4	x64	サイレント	acquirelin26x64.tar	Lin125_VN69_1	12.50	2.2.1
		z		acquirezlin26suse11.tar	Lin125_VN69_6		

## 2.2. インストール手順

### 2.2.1. サイレント形式のインストール

- ①対象サーバにユーザ ID : athene でログインします。  
pwd コマンドで athene のホームディレクトリであることを確認してください。
- ②「LANG=C ; export LANG」を入力します。
- ③「tar -xvf acquire\*\*\*.tar」を入力すると解凍処理が開始されます。  
\*\*\*に入る文字は OS 毎に異なります。
- ④「sh acqsetup」を入力すると、自動でインストールが実行されます。
- ⑤インストールログ確認  
athene のホームディレクトリに、「acqsetup.log」というインストール作業のログファイルが書き出されます。このファイルは、何か問題があった場合の調査に使用します。

Acquire のインストール作業は以上です。

Oracle データを収集する場合は 2.3.へ、Symfoware データを収集する場合は 2.4.へ進んでください。

## 2.3. Oracle データの収集設定

Acquire は Oracle Wallet を使用してリスナー経由で対象 Oracle データベースに接続し、V\$動的パフォーマンスビューから必要なデータを収集します。ここでは Oracle データを収集する為の設定について説明します。

### メモ！

Acquire は、自分専用の Oracle Wallet、tnsnames.ora、sqlnet.ora の設定を使用します。そのため、Oracle を使用する他のユーザの動作環境には影響を与えません。

### 2.3.1. Oracle 関連の環境変数のセット

Acquire 用ユーザで Oracle 関連のコマンドを実行するための環境変数をセットします。

- ①ユーザ ID : athene でログインし、athene のホームディレクトリであることを確認します。
- ②以下の環境変数をセットするコマンドを実行します。

```
ORACLE_SID=<SID 名>; export ORACLE_SID
ORACLE_HOME=<Oracle ホームのパス>; export ORACLE_HOME
PATH=$PATH:<Oracle ホームのパス>/bin; export PATH
```

2.3.2.へ進んでください。

### 2.3.2. Oracle Wallet 配置ディレクトリの作成

Acquire 用ユーザのホームディレクトリ下に、Oracle Wallet を配置するディレクトリを作成します。

- ①以下のコマンドを実行します。

```
mkdir /<Acquire ホームディレクトリ>/secure
chmod 700 /<Acquire ホームディレクトリ>/secure
```

(例)

```
ls -l /home/athene/secure
drwx-----. 2 athene users secure
```

2.3.3.へ進んでください。

### 2.3.3. Oracle Wallet の作成

「2.3.2.」で作成したディレクトリに Oracle Wallet を作成します。

- ①以下のコマンドを実行します。

```
mkstore -wrl <Oracle Wallet 配置ディレクトリ> -create
```

<Oracle Wallet 配置ディレクトリ> : 「2.3.2.」で作成したディレクトリ

(例)

```
mkstore -wrl /home/athene/secure -create
```

②パスワードの入力を求められますので、適切なパスワードを設定します。

Enter password:  
Enter password again:

ここでのパスワードは、Oracle Wallet を更新する際に必要となります。  
パスワードは、8 文字以上の長さで、数字または特殊文字、およびアルファベット文字を含める必要があります。

③Oracle Wallet が作成されていることを確認します。

(例)

```
ls -l /home/athene/secure
-rw----- 1 athene users  cwallet.sso
-rw----- 1 athene users  cwallet.sso.lck
-rw----- 1 athene users  ewallet.p12
-rw----- 1 athene users  ewallet.p12.lck
```

2.3.4.へ進んでください。

#### 2.3.4. Acquire 用ネットサービス名の設定

Acquire 用の Oracle ネットサービス名の設定を行います。

①テキストエディタ等を使用して、Acquire 用ユーザのホームディレクトリに「.tnsnames.ora」ファイルを作成します。

(例)

```
vi /home/athene/.tnsnames.ora
```

※ファイル名の先頭に必ず「.」を付加してください。

②「.tnsnames.ora」ファイルに、対象 Oracle データベースに接続可能な専用のネットサービス名を設定します。

```
<専用ネットサービス名> =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = <対象ホスト>)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = <対象インスタンス>)
  )
)
```

<専用ネットサービス名> : Acquire 用の Oracle ネットサービス名  
<対象ホスト> : 対象ホスト名または IP アドレス  
<対象データベース> : データ収集対象の Oracle インスタンスまたは PDB

#### メモ！

CDB 構成の複数 PDB のデータを収集する場合は、それぞれの PDB に対応するネットサービス名を設定します。

③「.tnsnames.ora」ファイルの編集完了後に、パーミッションを変更します。

```
chmod 700 /<Acquire ホームディレクトリ>/.tnsnames.ora
```

(例)

```
ls -la /home/athene/.tnsnames.ora
```

```
-rwx-----. 1 athene users .tnsnames.ora
```

※「.」から始まるファイルは「ls -la」コマンドで表示されます。

2.3.5.へ進んでください。

### 2.3.5. Acquire 用 sqlnet 環境の設定

Acquire 用の sqlnet 環境の設定を行います。

①テキストエディタ等を使用して、Acquire 用ユーザのホームディレクトリに「.sqlnet.ora」ファイルを作成します。

(例)

```
vi /home/athene/.sqlnet.ora
```

※ファイル名の先頭に必ず「.」を付加してください。

②「.sqlnet.ora」ファイルに、Oracle Wallet を使用するための下記設定を記述します。

```
WALLET_LOCATION =
(SOURCE =
  (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = <Oracle Wallet 配置ディレクトリ>)
    )
)

SQLNET.WALLET_OVERRIDE = TRUE
SSL_CLIENT_AUTHENTICATION = FALSE
SSL_VERSION = 0
```

<Oracle Wallet 配置ディレクトリ> : 「2.3.2.」で作成した Oracle Wallet 配置ディレクトリ

③「.sqlnet.ora」ファイルの編集完了後に、パーミッションを変更します。

```
chmod 700 /<Acquire ホームディレクトリ>/.sqlnet.ora
```

(例)

```
ls -la /home/athene/.sqlnet.ora
```

```
-rwx-----. 1 athene users .sqlnet.ora
```

※「.」から始まるファイルは「ls -la」コマンドで表示されます。

2.3.6.へ進んでください。

### 2.3.6. Acquire 用 Oracle ユーザ作成 SQL ファイルの編集

Acquire が Oracle データ取得時に使用する Oracle ユーザを作成するための create.sql を編集します。

- ①Acquire 用ユーザのホームディレクトリであることを確認します。
- ②「cd oracap」コマンドを実行します。
- ③「create.sql」をテキストエディタ等で開きます。

```
create user ops$metron identified externally temporary tablespace TEMP;
grant create session to ops$metron;
grant select on v_$database to ops$metron;
grant select on v_$datafile to ops$metron;
grant select on v_$dispatcher to ops$metron;
grant select on v_$event_name to ops$metron;
grant select on v_$filestat to ops$metron;
grant select on v_$latch to ops$metron;
grant select on v_$lock to ops$metron;
grant select on v_$librarycache to ops$metron;
grant select on v_$process to ops$metron;
grant select on v_$queue to ops$metron;
grant select on v_$rowcache to ops$metron;
grant select on v_$session to ops$metron;
grant select on v_$session_event to ops$metron;
grant select on v_$session_wait to ops$metron;
grant select on v_$sesstat to ops$metron;
grant select on v_$sga to ops$metron;
grant select on v_$sgastat to ops$metron;
grant select on v_$shared_pool_reserved to ops$metron;
grant select on v_$shared_server to ops$metron;
grant select on v_$sql to ops$metron;
grant select on v_$statname to ops$metron;
grant select on v_$system_event to ops$metron;
grant select on v_$sysstat to ops$metron;
grant select on v_$tablespace to ops$metron;
grant select on dba_free_space to ops$metron;
grant select on dba_data_files to ops$metron;
grant select on dba_segments to ops$metron;
grant select on dba_extents to ops$metron;
grant select on dba_objects to ops$metron;
grant select on v_$sort_segment to ops$metron;
grant select on dba_tablespaces to ops$metron;
grant select on dba_temp_files to ops$metron;
exit;
```



④1 行目の「create user 〜」を対象環境に合わせて編集します。

<非 CDB 構成>

変更前 :

```
create user ops$metron identified externally temporary tablespace TEMP;
```

変更後 :

```
create user <ユーザ名> identified by <パスワード> temporary tablespace <一時表領域>;
```

<ユーザ名> : Acquire 用 Oracle ユーザ名

<パスワード> : Acquire 用 Oracle ユーザに対するパスワード

<一時表領域> : Acquire 用 Oracle ユーザに対して割り当てる一時表領域

<CDB 構成-CDB>

変更前 :

```
create user ops$metron identified externally temporary tablespace TEMP;
```

変更後 :

```
create user <共通ユーザ名> identified by <パスワード>  
temporary tablespace <一時表領域> CONTAINER=ALL;
```

<共通ユーザ名> : Acquire 用 CDB/PDB 共通ユーザ名 (c##から始まる)

<パスワード> : Acquire 用 CDB/PDB 共通ユーザに対するパスワード

<一時表領域> : Acquire 用 CDB/PDB 共通ユーザに対して割り当てる一時表領域

**メモ!**

CDB 構成における CDB 環境のデータを取得する場合は、「c##」から始まる CDB/PDB 共通ユーザを作成する必要があります (例 : c##athene)。

CDB/PDB 共通ユーザは、CDB とすべての PDB に対してログイン可能です (各 PDB のデータも取得できます)。CREATE 文に「CONTAINER=ALL」オプションを指定します。

### <CDB 構成-PDB>

変更前 :

```
create user ops$metron identified externally temporary tablespace TEMP;
```

変更後 :

```
create user <PDB ユーザ名> identified by <パスワード>  
temporary tablespace <一時表領域> CONTAINER=CURRENT;
```

<PDB ユーザ名> : Acquire 用 PDB ローカルユーザ名 (例 : PDBATHENE)

<パスワード> : Acquire 用 PDB ローカルユーザに対するパスワード

<一時表領域> : Acquire 用 PDB ローカルユーザに対して割り当て一時表領域

#### **メモ !**

**CDB 構成における PDB 環境のみのデータを取得する場合は、PDB ローカルユーザを作成します。**

**PDB ローカルユーザは、対象 PDB に対してのみログイン可能です。**

**CREATE 文に「CONTAINER=CURRENT」オプションを指定します。**

⑤「grant ～～ to ops\$metron;」の行の「ops\$metron」を④で記述した Oracle ユーザに変更します。

### <非 CDB 構成>

変更前 :

```
grant create session to ops$metron;  
grant select on v_$database to ops$metron;  
～以下同様
```

変更後 :

```
grant create session to <ユーザ名>;  
grant select on v_$database to <ユーザ名>;  
～以下同様
```

<ユーザ名> : Acquire 用 Oracle ユーザ名

## &lt;CDB 構成-CDB&gt;

変更前 :

```
grant create session to ops$metron;  
grant select on v_$database to ops$metron;  
～以下同様
```

変更後 :

```
grant create session to <共通ユーザ名> CONTAINER=ALL;  
grant select on v_$database to <共通ユーザ名> CONTAINER=ALL;  
～以下同様
```

&lt;共通ユーザ名&gt; : Acquire 用 CDB/PDB 共通ユーザ名 (c##から始まる)

**メモ !**

CDB 構成-CDB の場合は、GRANT 文に「CONTAINER=ALL」オプションを指定します。

## &lt;CDB 構成-PDB&gt;

変更前 :

```
grant create session to ops$metron;  
grant select on v_$database to ops$metron;  
～以下同様
```

変更後 :

```
grant create session to <PDB ユーザ名> CONTAINER=CURRENT;  
grant select on v_$database to <PDB ユーザ名> CONTAINER=CURRENT;  
～以下同様
```

&lt;PDB ユーザ名&gt; : Acquire 用 PDB ローカルユーザ名

**メモ !**

CDB 構成-PDB の場合は、GRANT 文に「CONTAINER=CURRENT」オプションを指定します。

(例)

```
create user c##athene identified by password temporary tablespace temp CONTAINER=ALL;
grant create session to c##athene CONTAINER=ALL;
grant select on v_$database to c##athene CONTAINER=ALL;
grant select on v_$datafile to c##athene CONTAINER=ALL;
grant select on v_$dispatcher to c##athene CONTAINER=ALL;
grant select on v_$event_name to c##athene CONTAINER=ALL;
grant select on v_$filestat to c##athene CONTAINER=ALL;
grant select on v_$latch to c##athene CONTAINER=ALL;
grant select on v_$lock to c##athene CONTAINER=ALL;
grant select on v_$librarycache to c##athene CONTAINER=ALL;
grant select on v_$process to c##athene CONTAINER=ALL;
grant select on v_$queue to c##athene CONTAINER=ALL;
grant select on v_$rowcache to c##athene CONTAINER=ALL;
grant select on v_$session to c##athene CONTAINER=ALL;
grant select on v_$session_event to c##athene CONTAINER=ALL;
grant select on v_$session_wait to c##athene CONTAINER=ALL;
grant select on v_$sesstat to c##athene CONTAINER=ALL;
grant select on v_$sga to c##athene CONTAINER=ALL;
grant select on v_$sgastat to c##athene CONTAINER=ALL;
grant select on v_$shared_pool_reserved to c##athene CONTAINER=ALL;
grant select on v_$shared_server to c##athene CONTAINER=ALL;
grant select on v_$sql to c##athene CONTAINER=ALL;
grant select on v_$statname to c##athene CONTAINER=ALL;
grant select on v_$system_event to c##athene CONTAINER=ALL;
grant select on v_$sysstat to c##athene CONTAINER=ALL;
grant select on v_$tablespace to c##athene CONTAINER=ALL;
grant select on dba_free_space to c##athene CONTAINER=ALL;
grant select on dba_data_files to c##athene CONTAINER=ALL;
grant select on dba_segments to c##athene CONTAINER=ALL;
grant select on dba_extents to c##athene CONTAINER=ALL;
grant select on dba_objects to c##athene CONTAINER=ALL;
grant select on v_$sort_segment to c##athene CONTAINER=ALL;
grant select on dba_tablespaces to c##athene CONTAINER=ALL;
grant select on dba_temp_files to c##athene CONTAINER=ALL;
exit;
```

2.3.7.へ進んでください。

### 2.3.7. Acquire 用 Oracle ユーザ作成 SQL ファイルの実行

「2.3.6.」で編集した create.sql を実行し、Acquire 用 Oracle ユーザを作成します。

①対象の Oracle インスタンスまたは PDB にログインします。

＜非 CDB 構成/CDB 構成-CDB＞

「sqlplus」を実行します。

ユーザ ID、パスワードの入力要求が順次あります。

ユーザ「sys as sysdba」とパスワードを入力してください。

＜CDB 構成-PDB＞

「sqlplus」を実行します。

ユーザ ID、パスワードの入力要求が順次あります。

ユーザ「sys as sysdba」とパスワードを入力してください。

まず、CDB にログインした状態になります。

次に、「alter session set container = <PDB 名>;」を実行して、対象 PDB に切り替えます。

②「2.3.6.」で編集した create.sql を実行します。

```
sqlplus> @/<Acquire ホームディレクトリ>/oracap/create
```

(例)

@/home/athene/oracap/create

@の後に create.sql のフルパスを指定します。

```
ユーザが作成されました。
権限付与が成功しました。
権限付与が成功しました。
権限付与が成功しました。
権限付与が成功しました。
権限付与が成功しました。
権限付与が成功しました。
権限付与が成功しました。
権限付与が成功しました。
権限付与が成功しました。
権限付与が成功しました。
権限付与が成功しました。
権限付与が成功しました。
権限付与が成功しました。
権限付与が成功しました。
権限付与が成功しました。
権限付与が成功しました。
権限付与が成功しました。
権限付与が成功しました。
権限付与が成功しました。
```

権限付与が成功しました。  
権限付与が成功しました。  
権限付与が成功しました。  
権限付与が成功しました。  
権限付与が成功しました。  
権限付与が成功しました。  
権限付与が成功しました。  
権限付与が成功しました。

**メモ！**

対象環境によって SQL\*Plus から返るメッセージが日本語または英語になります。

**メモ！**

Oracle12c 以降では、パスワード認証タイプのユーザにおけるデフォルトのパスワード有効期限が「180 日」です。継続してデータ収集を行うために、Acquire 用 Oracle ユーザのパスワード有効期限が「UNLIMITED」になっている必要があります。SQL\*Plus から次の SELECT 文を実行することでパスワード有効期限を確認できます。

```
select u.username,p.resource_name,p.limit
  from dba_users u,dba_profiles p
 where u.profile = p.profile
       and u.username = '<ユーザ名>'
       and p.resource_name = 'PASSWORD_LIFE_TIME';
```

2.3.8.へ進んでください。

### 2.3.8. Oracle Wallet への認証情報の登録

Acquire が Oracle データ取得時に使用する Oracle ユーザ情報を Oracle Wallet に登録します。

①以下のコマンドを実行します。

```
mkstore -wrl <Oracle Wallet 配置ディレクトリ> -createCredential <ネットサービス名> <Oracle ユーザ>
<Oracle ユーザに対するパスワード>
```

Enter password:

<Oracle Wallet 配置ディレクトリ> : 「2.3.2. Oracle Wallet 配置ディレクトリの作成」のディレクトリ  
 <ネットサービス名> : 「2.3.4. Acquire 用ネットサービス名の設定」のネットサービス名  
 <Oracle ユーザ> : 「2.3.6. Acquire 用 Oracle ユーザ作成 SQL ファイルの編集」のユーザ  
 <Oracle ユーザに対するパスワード> : 「2.3.6. Acquire 用 Oracle ユーザ作成 SQL ファイルの編集」のユーザに対する  
 パスワード  
 Enter password: : 「2.3.3. Oracle Wallet の作成」の Oracle Wallet に対するパスワード

(例)

```
mkstore -wrl /home/athene/secure -createCredential ORCL athene password
mkstore -wrl /home/athene/secure -createCredential PDB1 pdb1athene password
```

2.3.9.へ進んでください。

### 2.3.9. Oracle への接続確認

対象 Oracle インスタンスまたは PDB にログインできるかどうかを確認します。

①以下のコマンドを実行します。

```
sqlplus /@<ネットサービス名>
```

<ネットサービス名> : 「2.3.4. Acquire 用ネットサービス名の設定」のネットサービス名

(例)

```
sqlplus /@ORCL
sqlplus /@PDB1
```

②SQL\*Plus のプロンプトが表示されたら check.sql を実行し、「OK」が表示されることを確認します。

```
sqlplus> @/<Acquire ホームディレクトリ>/oracap/check
```

(例)

```
@/home/athene/oracap/check
@の後に check.sql のフルパスを指定します。
```

```
v$database OK
v$datafile OK
v$dispatcher OK
v$event_name OK
v$filestat OK
v$latch OK
v$librarycache OK
v$lock OK
v$process OK
v$queue OK
v$rowcache OK
v$session OK
v$session_event OK
v$session_wait OK
v$sesstat OK
v$sga OK
v$sgastat OK
v$shared_pool_reserved OK
v$shared_server OK
v$sql OK
v$statname OK
v$system_event OK
v$sysstat OK
v$tablespace OK
dba_free_space OK
dba_data_files OK
dba_segments OK
dba_extents OK
dba_objects OK
v$sort_segment OK
dba_tablespaces OK
dba_temp_files OK
```

**メモ！**

- 1.非 CDB 構成の複数インスタンスのデータを収集する場合は、2.3.1.～2.3.9.の手順をインスタンス毎に実行してください。
- 2.CDB 構成の複数 PDB のデータを、各 PDB のローカルユーザを使用して収集する場合は、2.3.7.～2.3.9.の手順を PDB 毎に実行してください。



## 2.4. Symfoware データの収集設定

### 2.4.1. Symfoware データを収集する場合の設定

ここでは、Symfoware データを収集する為の設定について説明します。

- ①ユーザ ID : athene でログインします。
- ②athene のホームディレクトリであることを確認します。
- ③「sh scr/symfw.inst.sh」を実行します。「Enter」を入力して先に進めてください。

This is the installation script for the Symfoware option of the Metron data capture agent.

Checks will now be performed on the current capture environment to ensure that there are no problems.

Working . . . . .

The environment seems to be OK so far.

Proceeding

Create symfoware directory if it's not there already

Press return to continue

- ④次の画面が表れたら、「Enter」を入力して先に進めてください。

Two environment variables are required for Symfoware capture agent as follows:-

The path to the symfoware bin directory

The path to the symfoware lib directory

You may change the details after this installation by running this symfw.inst.sh script again

Press return to continue

⑤次の画面では、Symfoware の/bin ディレクトリパスを入力します。

通常、このパスには「/opt/FJSVrdb2b/bin」を指定しますが、念の為 Symfoware インストール先ディレクトリを確認してください。

Please enter the path to the symfoware bin directory now

⑥次の画面では、Symfoware の/lib ディレクトリパスを入力します。

通常、このパスには「/opt/FJSVrdb2b/lib」を指定しますが、念の為 Symfoware インストール先ディレクトリを確認してください。

Please enter the path to the symfoware lib directory now

⑦次の画面が表れたら設定は終了です。

Data accepted

Extract the actual symfoware capture agent to the /export/home/athene/bin directory  
Thank youThe set-up phase for the Symfoware capture mechanism is complete

Exiting

Done

Symfoware の設定は以上です。

## 第3章 アンインストール

ここでは、Acquire のアンインストール手順を説明します。

### 3.1. Acquire のアンインストール手順

#### 3.1.1. 削除手順

- ①ユーザ ID : athene でログインします。
- ②pwd コマンドで athene のホームディレクトリであることを確認します。
- ③sh bin/APMstopit を入力します。最後のインターバルの処理を終えてからデータ収集が停止します。
- ④crontab -r を入力して、cron の登録内容を削除します。
- ⑤crontab -l で登録内容が削除されたことを確認します。
- ⑥athene のホームディレクトリ内にあるファイルやディレクトリを削除して、ホームディレクトリを空の状態にします。
- ⑦/var/adm または /var/account にある Metron.save.d ディレクトリを削除します。
- ⑧「root」にスイッチユーザするか、ログインし直します。
- ⑨cron.allow を使用している場合は、中に書かれている athene を削除します。
- ⑩設定した Acquire 用ユーザ ID : athene を削除します。
- ⑪Oracle データを収集していた場合、設定した Acquire 用 Oracle ユーザを削除します。
- ⑫Acquire 停止後、課金機能を使用しない場合には、以下の手順に従って課金機能を停止します。
  - ・下記コマンドを実行します。

/usr/lib/acct/accton	(AIX、Solaris、HP-UX)
/usr/sbin/accton off	(RHEL7/8/9、OracleLinux7、SLES12/15)
/usr/sbin/accton	(その他の Linux)
  - ・pacct の自動削除に関する cron の記述を削除します。
  - ・システムリブート時に課金機能を自動起動するように設定している場合、Acquire アンインストール後に不要であれば設定を無効にします。