



仮想化
友の会

第2回

濃いバナ Xen API

～ Xen APIで実現する仮想マシン制御 ～

仮想化友の会

平初 <htaira@pantora.net>



○○○から

Xen APIを叩き、

Xenを○○○する



- 正式名称はXen Management API
- Xenを管理するための制御API
- XML-RPCベースで通信を行う
- Xen 3.1.0から実装された
- Fedora 7、SLES10 SPI(一部バックポートされた)で利用可能
- クライアント側の環境を選ばない



- 例えば、こんな感じ (Python)

```
#!/usr/bin/python

import xmlrpclib
xend = xmlrpclib.Server("http://127.0.0.1:9363")

session = xend.session.login_with_password("root", "password")['Value']

for method in xend.host.list_methods(session)['Value']:
    print method

xend.session.logout(session)
```

※1 分かりやすくするために例外処理を省いています。

※2 SLES10 SPIのXen APIではsession.host.list_methods()が実装されていません。



- XMLをやりとりする
- SOAPより簡単な仕組み
- blogのAPIでよく使われている
- 通信はSSL暗号化が可能
- NATやWeb Proxyも通過可能
- クライアント側の環境を選ばない

詳しくは → <http://ja.wikipedia.org/wiki/XML-RPC>



- Xenを制御する手順
 - サーバーを指定する
 - ログインし、セッションを確率する
 - 制御関数を呼ぶ（繰り返し）
 - ログアウトし、セッションを開放する



- サーバーを指定する

Python

```
xend = xmlrpclib.Server("http://127.0.0.1:9363")
```

C#

```
public interface IXenAPI : IXmlRpcProxy  
{  
    ... 省略 ...  
}  
  
... 省略 ...  
  
IXenAPI = XmlRpcProxyGen.Create<IXenAPI>();  
xend.Url = "http://127.0.0.1:9363";
```



- ログインし、セッションを確率する

Python

```
session = xend.session.login_with_password("root","password")['Value']
```

C#

```
public interface IXenAPI : IXmlRpcProxy
{
    [XmlRpcMethod("session.login_with_password")]
    string LoginWithPassword(string username, string password);
    ... 省略 ...
}

... 省略 ...

string session = xend.LoginWithPassword("root","password");
```



- 制御関数を呼ぶ

Python

```
methods = xend.host.list_methods(session)
```

C#

```
public interface IXenAPI : IXmlRpcProxy
{
    ... 省略 ...

    [XmlRpcMethod("host.list_methods")]
    string[] ListMethods(string session_id);
    ... 省略 ...
}

... 省略 ...

string[] methods = xend.ListMethods(session);
```



- ログアウトし、セッションを開放する

Python

```
xend.logout(session)
```

C#

```
public interface IXenAPI : IXmlRpcProxy
{
    ... 省略 ...

    [XmlRpcMethod("session.logout")]
    void Logout(string session_id);
}

... 省略 ...

xend.Logout(session);
```



- C#用XML-RPCライブラリ
「XML-RPC.net」を利用する
- C#でXenを制御できる
- Windowsアプリケーションから
Xenを制御できる



なぜC#かって？

だって、**C#**だと

みんな**分かる**でしょ？



オレは

Python
が好き(笑)



でも、そんなの関係ねえ！



- C#でXML-RPCを書くと汚い
ってかキモイ。キモすぎ。
- 今後、誰か悩むだろう。
- ライブラリにしてしまえ～。



デキちゃいました。



C#用 Xen APIライブラリ



- 実装状況
 - 仮想マシンの起動、終了、再起動、
レジューム、サスペンド、停止、
CPU使用量、メモリ使用量
 - Xenのdmesg、Xenのログ情報の取得



```
GA コマンド プロンプト

C:\Project\XenAPI>xm.exe connect http://root:password@172.16.31.1:9363
XML_SESSION_URL: http://172.16.31.1:9363/
XML_SESSION_ID: a55aed8b-08e3-2acb-886d-3d06db94eb43

C:\Project\XenAPI>xm.exe list
Name                ID      Mem    VCPUs  State
Domain-0            0       5460   4      Running
winxp                -1       0      0      Halted
vista                -1       0      0      Halted
fedora6              -1       0      0      Halted
winsrv2003          -1       0      0      Halted

C:\Project\XenAPI>xm.exe start vista

C:\Project\XenAPI>xm.exe list
Name                ID      Mem    VCPUs  State
Domain-0            0       5460   4      Running
vista                11      2055   1      Running
winxp                -1       0      0      Halted
fedora6              -1       0      0      Halted
winsrv2003          -1       0      0      Halted

C:\Project\XenAPI>xm.exe pause vista

C:\Project\XenAPI>xm.exe list
Name                ID      Mem    VCPUs  State
Domain-0            0       5460   4      Running
vista                11      2055   2      Paused
winxp                -1       0      0      Halted
fedora6              -1       0      0      Halted
winsrv2003          -1       0      0      Halted

C:\Project\XenAPI>xm.exe shutdown vista

C:\Project\XenAPI>xm.exe disconnect

C:\Project\XenAPI>
```



HP ProLiant ML115



HP Directplusで買えます。
15,000円(税別)



Windows上から 仮想マシンを管理

- xendへの接続
- リスト表示
- 起動
- 停止
- ポーズ
- アンポーズ
- dmesgの確認
- xendログの確認



仮想化
友の会

仮想マシン制御の未来



レンダリングファームの
有効利用



アプリケーションの
隔離デバッグの自動化



分散コンパイル環境の最適化



リソースプールの
ユーティリティ課金





仮想化 友の会