

## Java Bridgeを利用した他言語によるデータロード & プロットデモ

阿部, 修司  
九州大学国際宇宙天気科学・教育センター

IUGONETプロジェクトチーム  
九州大学国際宇宙天気科学・教育センター

<https://hdl.handle.net/2324/27196>

---

出版情報 : 2013-08-21  
バージョン :  
権利関係 :



IUGONET

Metadata DB for Upper Atmosphere

超高層大気長期変動の全球地上ネットワーク観測・研究  
Inter-university Upper atmosphere Global Observation NETwork

# Java Bridgeを利用した他言語による データロード & プロットデモ

担当: 阿部 (九大ICSWSE)

# Outline

1. Java bridgeとは
2. JUDASとは
3. Java bridgeを使う

# Purpose

- Java bridgeとその仕組みを知る
- 他の言語からJavaクラスを呼び出してIUGONET機関の取り扱うデータをプロットできるようになる

## Java bridgeとは

ソフトウェア開発者が2つのプログラミング言語を統合するために、これらを(相互に)使用できる構造を提供するコンポーネントがある。このコンポーネントを bridge と呼ぶ。  
われわれが作成している データ読み出し、解析、描画用bridge は Java で書かれているため、特に Java bridge と呼んでいる。

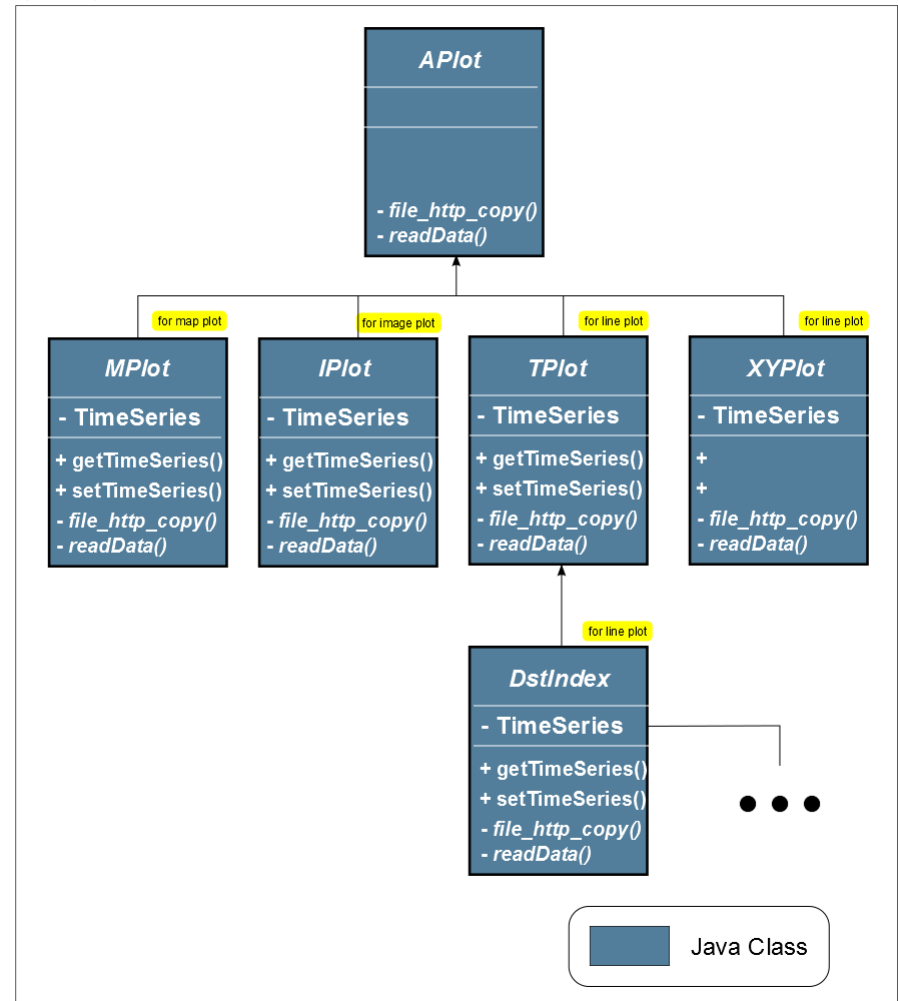
## なぜ Java で提供するのか

- 特定のプラットフォームに依存しなくなる
- プログラムの保守が容易になる
- 他言語のコード資産との接続が容易になる

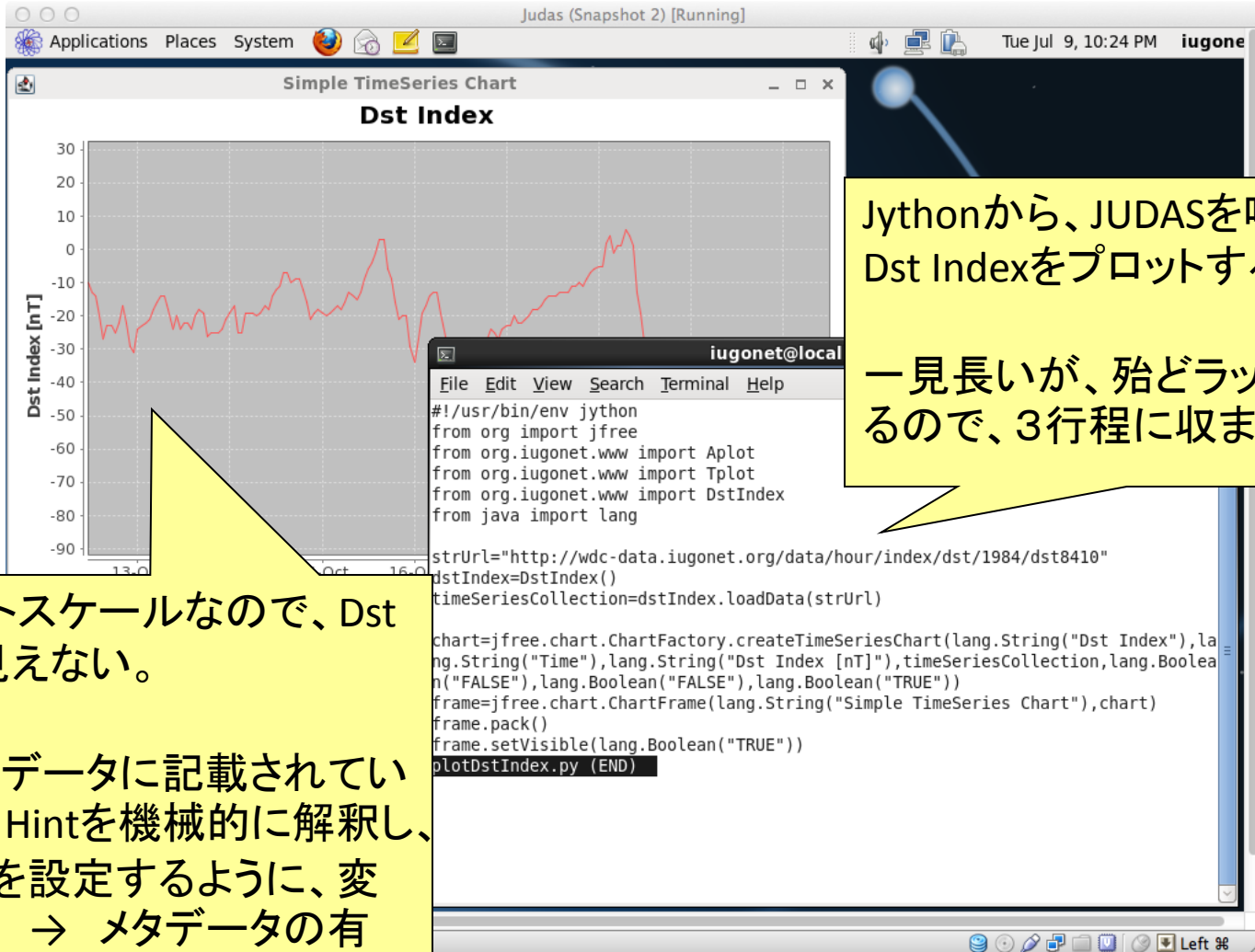
# JUDAS(Java-based UDAS)とは？

- Java言語で書かれたライブラリ群。
- 観測データのダウンロード、プロット、解析が可能。
- 単独のダウンロードツールとしても利用可能。
- 他言語(e.g., Jython, JRuby, MATLAB, IDL(確認中))から呼び出すことが可能。
- Java Web Startを用いた簡単インストール&更新。
- CUI(Jython, JRuby)もGUIも準備中。
- Metadata DBのQuickLookへの利用。(既に、DstIndexはQuickLook組み込み済み)

Deployment Diagram 1  
(Judas)



# JUDASのサンプル



Jythonから、JUDASを呼び出して  
Dst Indexをプロットするサンプル。

一見長いですが、殆どラッピング出来るので、3行程に収まる。

縦軸がオートスケールなので、Dst Indexらしく見えない。

現在は、メタデータに記載されているRendering Hintを機械的に解釈し、適切な縦軸を設定するように、変更している。→ メタデータの有機的な利用！

# Java bridgeを使う

## Judasを入手しよう

### 推奨手順は

1. Scientific Linux6 64bit環境(VMでも良い)を準備する
2. 作業アカウント“iugonet”を作成する
3. Github からJudas をダウンロードして展開する
4. Ant をインストールする
5. Judas付属の“build.xml”を実行して環境を構築する  
(開発環境eclipseのインストールを含む)
6. Eclipse上でファイルを書き直し、パッケージを作成する

なのですが、今回は時間がないので上記そのままの実習は割愛。

3. Githubからのファイル取得をやってみましょう。

Eclipse: IBMによって開発された統合開発環境。Javaで記述され、オープンソースで公開。Javaのみならず多くのプログラミング言語に対応。  
**IUGONETのメタデータ記述推奨環境**でもある。

# Java bridgeを使う

Judasを入手しよう

IUGONET開発プロダクトのほとんどは  
GitHub上で公開されています！

<https://github.com/iugonet/>

**GitHub** Search or type a command

Explore Features Enterprise Blog Sign up Sign in

Contributions **Repositories** Public Activity Follow

### Popular repositories

Repository	Stars
<a href="#">DSpaceSemanticSearchTestE...</a>	1 ★
<a href="#">iugonetAssociativeSearch</a>	1 ★
<a href="#">iugonetMetadataDB</a>	1 ★
<a href="#">DE1Digitizer</a>	1 ★
<a href="#">iugonetMetadataEditor</a>	1 ★

### Public contributions

Summary of Pull Requests, issues opened and commits. [Learn more.](#)

Less More

Category	Count	Period
Total	239	Aug 17 2012 - Aug 17 2013
Streak	6 days	June 01 - June 06
Current Streak	0 days	Rock - Hard Place



# Java bridgeを使う

Judasを入手しよう

<https://github.com/iugonet/>

The screenshot shows the GitHub profile page for user **Atsuki Shinbori iugonet**. The profile includes a red pixelated avatar, a bio with contact information, and statistics: 0 followers, 5 starred, and 0 following. The repository list is filtered to show only repositories owned by iugonet. The repositories listed are:

- iugonetMetadataDB** (Java, 1 star, 0 forks, last updated 23 minutes ago)
- Udas** (Prolog, 0 stars, 0 forks, last updated 5 hours ago)
- UdasCheck** (Prolog, 0 stars, 0 forks, last updated 17 hours ago)
- Judas** (Java, 0 stars, 0 forks, last updated 10 days ago)

Annotations on the image:

- A red box highlights the repository name **Judas** with the text "ここをクリック" (click here).
- A yellow arrow points downwards with the text "スクロール" (scroll).
- A search bar above the repository list contains the text "Filter iugonet's repositories...".

# Java bridgeを使う

Judasを入手しよう

<https://github.com/iugonet/>

92 commits   1 branch   0 releases   1 contributor

branch: master Judas

Fixed

Yukinobu KOYAMA authored 11 days ago

- conf
- workspace Fixed
- .gitignore Fixed
- README.md Fixed
- build.properties Fixed
- build.xml Fixed

README.md

**Test**

Code

Issues 0

Pull Requests

Wiki

Pulse

Graphs

Network

HTTPS clone URL

<https://github.com>

You can clone with HTTPS, Subversion, and other methods.

Clone in Desktop

Download ZIP

スクロール

Git環境がある人は、このURLで git clone でもよし

ここをクリックするとzipがダウンロードできる

# Java bridgeを使う

Judasを入手しよう

<https://github.com/iugonet/>

	名前	更新日時	種類	サイズ
Fixed	gnuplot	2013/08/18 12:32	ファイル フォル...	
Yukinobu KOYAMA authored 11 days ago	JRuby	2013/08/18 12:32	ファイル フォル...	
..	Jython	2013/08/18 12:32	ファイル フォル...	
JRuby	Mathematica	2013/08/18 12:32	ファイル フォル...	
Jython	MATLAB	2013/08/18 12:32	ファイル フォル...	
MATLAB	Maxima	2013/08/18 12:32	ファイル フォル...	
Mathematica	Octave	2013/08/18 12:32	ファイル フォル...	
Maxima	R	2013/08/18 12:32	ファイル フォル...	
Octave	bashrc	2013/08/18 12:32	ファイル	1 KB
R				
gnuplot				
bashrc				

ダウンロードしたアーカイブをJudasディレクトリに展開すると。。。  
**Judas/workspace/Plugins/bridges**  
に各言語へのjava bridgeが準備されている

# Java bridgeを使う

## Judasを入手しよう

1. 本実習用のビルド済jarファイルを以下からダウンロードする  
<http://magdas2.serc.kyushu-u.ac.jp/bridges-demo.zip>
2. 適当な場所に展開すると、bridges-demoというディレクトリができる

名前	更新日時	種類	サイズ
lib	2013/08/18 23:00	ファイル フォル...	
plugins	2013/08/18 23:00	ファイル フォル...	

### Libの中身は

名前
external
Aplot.jar
Tplot.jar

### 名前

jcommon-1.0.18.jar
jfreechart-1.0.15.jar

### pluginsの中身は

### 名前

DstIndex.jar
--------------

Jython:プログラミング言語PythonをJavaで実装したもの。  
あらゆるJavaクラスをシームレスにインポートできるのが特徴のひとつ。

# Java bridgeを使う

サンプル実行用に他言語(ここではJython)をインストールする

Linuxの場合:

各ディストリビューションに付属のパッケージマネージャを使うと楽  
(例えば、SLでは `# yum install jython`)  
また、Judas内の `build.xml`を実行すればインストールできる

Mac/Winの場合(LinuxでもOK):

1. <http://www.jython.org/> にアクセス



ここをクリック

# Java bridgeを使う

サンプル用に他言語(ここではJython)をインストールする

## 2. Downloadから“Download Jython 2.5.3” をクリック

**ABOUT**

- Welcome
- Download
- Installation
- Mailing Lists
- License
- Foundation
- Archived Sites
- Acknowledgments

**Downloads**

The most current stable release of Jython is 2.5.3. For production purposes, please use this version.

The 2.5.4rc1 release candidate is now available for use. This release fixes numerous bugs since the 2.5.3 release of Jython.

The Jython 2.7beta1 release is also available for download. Jython 2.7b1 brings us up to language level compatibility with the 2.7 version of CPython. We have focused largely on CPython compatibility, and so this release of Jython can run more pure Python apps than any previous release.

Please use the link below to download the Java installer. Once downloaded, please double-click on the JAR file to start the installation process. You may also want to read the [Installation instructions](#) or the [Release Notes](#).

**Jython 2.5.3**

[Download Jython 2.5.3](#)

**Twitter Updates**

[Follow Jython](#)

ここをクリック

## 3. Central Repository から“jython-installer”の“jar”をクリック

The Central Repository

SEARCH | ADVANCED SEARCH | BROWSE | QUICK STATS

g"org.python" AND v:"2.5.3"

SEARCH

New: App Scan | Advanced Search | API Guide | Help

Search Results < 1 > displaying 1 to 3 of 3

Groupid	Artifactid	Version	Updated	Download
<a href="#">org.python</a>	<a href="#">jython-standalone</a>	<a href="#">2.5.3</a>	14-Aug-2012	<a href="#">pom</a> <a href="#">jar</a> <a href="#">javadoc.jar</a> <a href="#">sources.jar</a>
<a href="#">org.python</a>	<a href="#">jython</a>	<a href="#">2.5.3</a>	14-Aug-2012	<a href="#">pom</a> <a href="#">jar</a> <a href="#">javadoc.jar</a> <a href="#">sources.jar</a>
<a href="#">org.python</a>	<a href="#">jython-installer</a>	<a href="#">2.5.3</a>	14-Aug-2012	<a href="#">pom</a> <a href="#">jar</a> <a href="#">javadoc.jar</a> <a href="#">sources.jar</a>

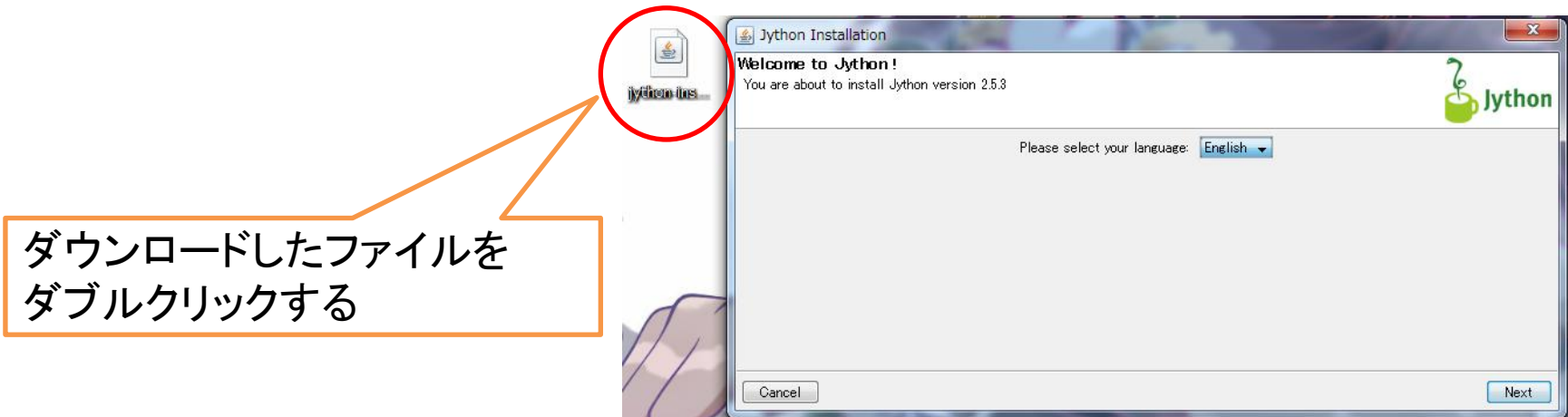
< 1 > displaying 1 to 3 of 3

ここをクリック

# Java bridgeを使う

サンプル用に他言語(ここではJython)をインストールする

## 4. ダウンロードしたjarファイルを実行する



または、コマンドライン(ターミナル)上で  
**# java -jar jython-installer-2.5.3.jar**  
を実行

あとはインストーラーの指示に従う

# Java bridgeを使う

サンプル用に他言語(ここではJython)をインストールする

## 5. Jythonを実行する

Windows:

コマンドプロンプト上で

>C:¥jython2.5.3¥bin¥jython

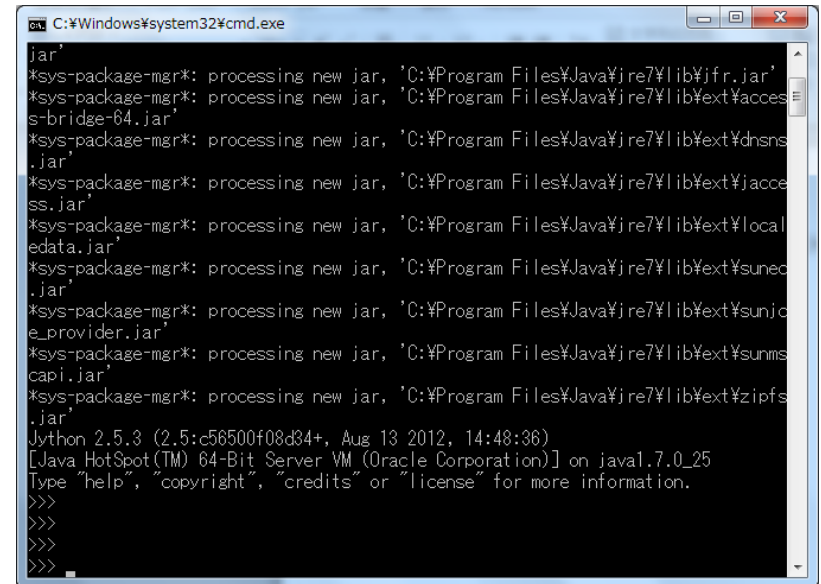
または

C:¥jython2.5.3¥bin¥jython.bat

をダブルクリックでもよい



```
iugonet@devel0:~$ jython
Jython 2.2.1 on java1.7.0_25
Type "copyright", "credits" or "license" for more information.
>>>
```



```
C:\Windows\system32\cmd.exe
jar
*sys-package-mgr*: processing new jar, 'C:\Program Files\Java\jre7\lib\jfr.jar'
*sys-package-mgr*: processing new jar, 'C:\Program Files\Java\jre7\lib\ext\access-bridge-64.jar'
*sys-package-mgr*: processing new jar, 'C:\Program Files\Java\jre7\lib\ext\dnsns.jar'
*sys-package-mgr*: processing new jar, 'C:\Program Files\Java\jre7\lib\ext\jaccess.jar'
*sys-package-mgr*: processing new jar, 'C:\Program Files\Java\jre7\lib\ext\localisedata.jar'
*sys-package-mgr*: processing new jar, 'C:\Program Files\Java\jre7\lib\ext\sunec.jar'
*sys-package-mgr*: processing new jar, 'C:\Program Files\Java\jre7\lib\ext\sunec_e_provider.jar'
*sys-package-mgr*: processing new jar, 'C:\Program Files\Java\jre7\lib\ext\sunmscapi.jar'
*sys-package-mgr*: processing new jar, 'C:\Program Files\Java\jre7\lib\ext\zipfs.jar'
Jython 2.5.3 (2.5:c56500f08d34+, Aug 13 2012, 14:48:36)
[Java HotSpot(TM) 64-Bit Server VM (Oracle Corporation)] on java1.7.0_25
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>>
>>>
```

Linux:

\$ jython

などで起動する



# Java bridgeを使う

Jythonで実際にデータ読み込み、図を描いてみる

1. Bridgesから、`plotDstIndex2.py` を `bridges-demo` ディレクトリの中にコピーする

2. `plotDstIndex2.py` を書き直す

```
#!/usr/bin/env jython
```

Jythonの場所を指定

```
import sys
```

```
# Add jar path
```

```
sys.path.append("/home/iugonet/Judas/lib/Tplot.jar")
```

```
sys.path.append("/home/iugonet/Judas/lib/Aplot.jar")
```

```
sys.path.append("/home/iugonet/Judas/lib/external/jcommon-1.0.17.jar")
```

```
sys.path.append("/home/iugonet/Judas/lib/external/jfreechart-1.0.14.jar")
```

```
sys.path.append("/home/iugonet/Judas/plugins/DstIndex.jar")
```

データ読み込み、描画に必要なライブラリの位置を指定。ライブラリ名にも注意！  
Windowsでもディレクトリの区切りは「/」を使う

```
timeSeriesCollection=dstIndex.loadData(strUrl)
```

loadData->loadに変更

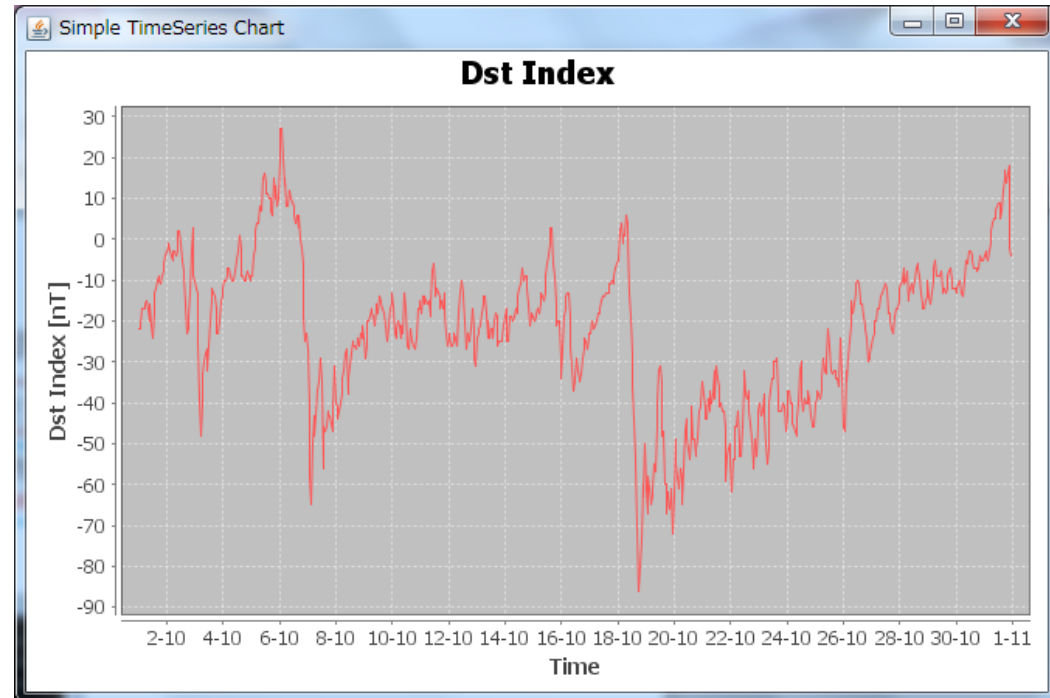
# Java bridgeを使う

Jythonで実際にデータ読み込み、図を描いてみる

## 3. 書き直したファイルを実行する

Win上での例: コマンドプロンプトで  
> c:¥jython2.5.3¥jython plotDstIndex2.py

```
コマンド プロンプト - c:¥jython2.5.3¥bin¥jython plotDstIndex2.py
DST8410*20 X219 000-063-049-055-058-061-056-058-065-056-047-044-049-051-054-048
-041-049-049-053-051-049-042-041-040-051
DST8410*21 X219 000-035-038-041-044-044-039-042-037-040-038-032-039-031-034-036
-041-040-042-042-046-059-053-051-050-041
DST8410*22 X219 000-055-062-054-054-046-046-042-050-053-053-046-036-032-038-039
-037-042-045-049-055-056-049-053-046-047
DST8410*23 X219 000-042-040-043-042-038-043-050-055-051-040-037-034-034-030-030
-029-038-042-042-042-040-040-041-047-040
DST8410*24 X219 000-043-037-037-040-040-045-046-047-048-044-040-032-030-039-042
-039-040-040-037-039-043-045-046-046-041
DST8410*25 X219 000-040-039-037-038-039-037-034-029-032-033-029-022-026-032-032
-033-032-032-034-034-036-025-024-033-033
DST8410*26 X219 000-046-047-032-035-029-025-021-015-018-016-013-011-010-011-015
-016-016-019-021-022-024-030-029-028-023
DST8410*27 X219 000-026-024-023-020-020-017-017-017-014-013-012-012-011-012-018
-018-020-022-023-019-019-017-017-015-018
DST8410*28 X219 000-012-011-010-007-011-010-008-015-014-012-011-013-012-010-007
-006-007-009-013-016-017-017-015-014-012
DST8410*29 X219 000-010-014-016-013-010-007-005-009-009-009-009-008-012-013-012
-013-012-009-008-007-010-012-012-012-010
DST8410*30 X219 000-013-012-011-010-013-013-014-010-005-005-006-003-003-004-005
-007-007-007-008-007-004-005-005-005-008
DST8410*31 X219 000-004-003-004-005-003-001 002 004 005 005 007 008 009 009 005
007 012 015 017 014 016 018-002-004 005
```



# Java bridgeを使う

プログラムの構成を理解して、  
他言語でのbridge作成に挑戦しよう

```
#!/usr/bin/env jython
import sys

# Add jar path
sys.path.append("/home/iugonet/Judas/lib/Tplot.jar")
sys.path.append("/home/iugonet/Judas/lib/Aplot.jar")
sys.path.append("/home/iugonet/Judas/lib/external/jcommon-1.0.17.jar")
sys.path.append("/home/iugonet/Judas/lib/external/jfreechart-1.0.14.jar")
sys.path.append("/home/iugonet/Judas/plugins/DstIndex.jar")

from org import jfree
from org.iugonet.www import Aplot
from org.iugonet.www import Tplot
from org.iugonet.www import DstIndex
from java import lang
```

初期化部:

Jarファイルへのパス設定や  
クラスのインポートなどをおこなう

```
strUrl="http://wdc-data.iugonet.org/data/hour/index/dst/1984/dst8410"
dstIndex=DstIndex()
timeSeriesCollection=dstIndex.load(strUrl)
```

データ読み込み部:

指定されたURLからデータを読み込む

```
chart=jfree.chart.ChartFactory.createTimeSeriesChart(lang.String("Dst Index"),lang.String("Time"),lang.String("Dst Index [nT]"),timeSeriesCollection,lang.Boolean("FALSE"),lang.Boolean("FALSE"),lang.Boolean("TRUE"))
frame=jfree.chart.ChartFrame(lang.String("Simple TimeSeries Chart"),chart)
frame.pack()
frame.setVisible(lang.Boolean("TRUE"))
```

データ解析、描画部:

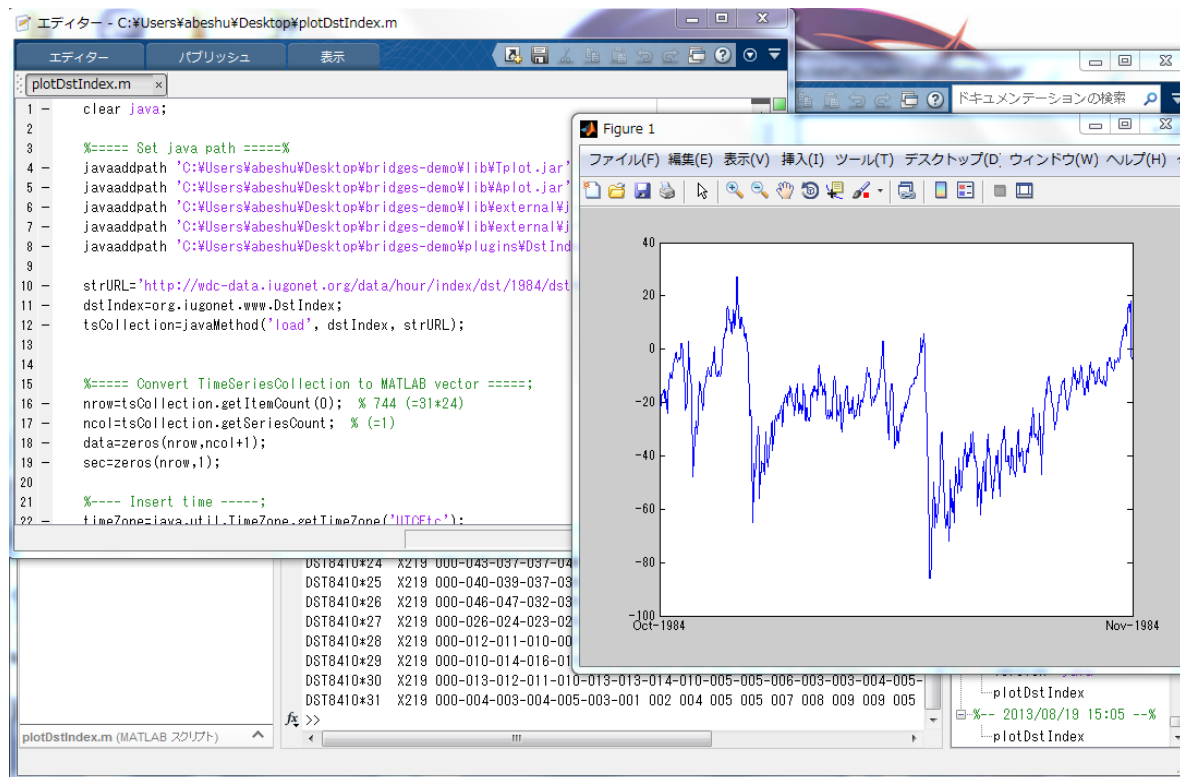
データ解析や描画をおこなう。ここを別言語に置換することも可能 (Java bridgeで読み込んでMATLABに描かせる、など)

# MATLAB/Java ブリッジ

## 利用のポイント

MATLABが使うJava、とjarファイルをコンパイルしたJavaのバージョンを合わせる  
どちらかで対応できる

1. Jar作成のJavacをMATLABが使うJavaのバージョンにする
2. 環境変数 MATLAB\_JAVA でMATLABが使うJavaをjarのJavaバージョンにする



# Appendix 1 JDK7 を入れる

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

1. Acceptを選ぶ

**Java SE Development Kit 7u25**  
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement  Decline License Agreement

Product / File Description	File Size	Download
Linux x86	80.38 MB	<a href="#">jdk-7u25-linux-i586.rpm</a>
Linux x86	93.12 MB	<a href="#">jdk-7u25-linux-i586.tar.gz</a>
Linux x64	81.46 MB	<a href="#">jdk-7u25-linux-x64.rpm</a>
Linux x64	91.85 MB	<a href="#">jdk-7u25-linux-x64.tar.gz</a>
Mac OS X x64	144.43 MB	<a href="#">jdk-7u25-macosx-x64.dmg</a>
Solaris x86 (SVR4 package)	136.02 MB	<a href="#">jdk-7u25-solaris-i586.tar.Z</a>
Solaris x86	92.22 MB	<a href="#">jdk-7u25-solaris-i586.tar.gz</a>
Solaris x64 (SVR4 package)	22.77 MB	<a href="#">jdk-7u25-solaris-x64.tar.Z</a>
Solaris x64	15.09 MB	<a href="#">jdk-7u25-solaris-x64.tar.gz</a>
Solaris SPARC (SVR4 package)	136.16 MB	<a href="#">jdk-7u25-solaris-sparc.tar.Z</a>
Solaris SPARC	95.5 MB	<a href="#">jdk-7u25-solaris-sparc.tar.gz</a>
Solaris SPARC 64-bit (SVR4 package)	23.05 MB	<a href="#">jdk-7u25-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	17.67 MB	<a href="#">jdk-7u25-solaris-sparcv9.tar.gz</a>
Windows x86	89.09 MB	<a href="#">jdk-7u25-windows-i586.exe</a>
Windows x64	90.66 MB	<a href="#">jdk-7u25-windows-x64.exe</a>

2. 適切なプロダクトをクリック

※1 Win/Linuxの場合、JREでも良い

(MacのJREが提供されていないので、本AppendixはJDKとした)

※2 Linuxの場合、各ディストリビューション提供のOpenJava7でも良い