
どこが使える? Java SE 6

横河電機
櫻庭 祐一



はじめに

- Java SE 6 Top 10!
 - Top 10 選考委員会 (委員長 櫻庭) による
厳正な選考
 - 選考のポイント
 - すぐに使える 単に SE 6 を使うだけでも恩恵がある
 - はまれば納得 特定の用途にピッタリの機能
 - じわじわと効く 使い方は未知数 いろいろな応用が可能





はじめに

• Top 10 候補機能

VM	GC Parallel Companion GC Concurrent Mark Sweep Collector Enhancement Ergonomics Enhancement Lock の効率化 Escape Analysis Classfile Specification Update (Verification) 複数Agent DTrace OutOfMemoryHandling		jhat jmap, jinfo
Core	ServiceLoader Deque NavigableSet/Map Arrays.copyOf	XML	StAX JAXB 2.0
I/O	Diskの空きスペース パーミッションの変更 Console	Web Services	JAX-WS 2.0 JAF Common Annotations Web Services Metadata
Net	NetworkInterface CookieManager	I18N	Local Sensitive Service API Unicode正規化 和暦 ResourceBundle enhancement Internationalized Domain Name
DB	JDBC 4.0 Derby	Java2D	OpenGL Single-thread pipeline JOGLとの統合 Direct3D効率化 文字表示の品質向上 (LCD ClearText) 描画精度向上 float -> double
Security	XML Digital Signature 楕円曲線暗号	Image I/O	GIImageWriter
EoD	Scripting Compiler API Pluggable Annotation Processor	AWT	New Modal Model Splash Screen System Tray Desktop class
M&M	MXBean自作 On Demand Attach jconsole Enhancement jconsole API Attach API @DescriptorKey 複合要素のモニタ Lockに関する情報収集	Swing	Vista & GTK L&F GroupLayout BaseLine Layout Drag & Drop Improve ウィンドウ単位のDouble Buffering JTable sort & filter SwingWorker JTextComponent印刷サポート JTabbedPaneのタブにJComponentを貼る
		Deployment	JWS enhancement





10位 GroupLayout

- すぐ使える
- はまれば納得 ◎
- じわじわと効く





10位 GroupLayout

- どんな機能?
 - Swing コンポーネントのレイアウト
かなり強力
 - もともと NetBeans の Matisse で使われていた
- どこに使うの?
 - 縦横ともに、そろえたいときなど
 - GridbagLayout, SpringLayout より使いやすい





10位 GroupLayout

- どうやって使うの？

```
JPanel panel = ...;  
GroupLayout layout = new GroupLayout(panel);  
panel.setLayout(layout);
```

```
layout.setAutoCreateGaps(true);  
layout.setAutoCreateContainerGaps(true);
```

```
GroupLayout.SequentialGroup hGroup = layout.createSequentialGroup();  
hGroup.addGroup(layout.createParallelGroup().  
    addComponent(label1).addComponent(label2));  
hGroup.addGroup(layout.createParallelGroup().  
    addComponent(field1).addComponent(field2));  
layout.setHorizontalGroup(hGroup);
```

```
GroupLayout.SequentialGroup vGroup = layout.createSequentialGroup();  
vGroup.addGroup(layout.createParallelGroup(Alignment.BASELINE).  
    addComponent(label1).addComponent(field1));  
vGroup.addGroup(layout.createParallelGroup(Alignment.BASELINE).  
    addComponent(label2).addComponent(field2));  
layout.setVerticalGroup(vGroup);
```





9位 JOGL と Java2D の統合

- すぐ使える
- はまれば納得
- じわじわと効く ◎





9位 JOGL と Java2D の統合

- どんな機能?
 - OpenGL のパイプラインの扱いを JOGL と統合
 - JOGL から Java2D を扱える
 - 使いこなすのは難しい
- どこに使うの?
 - カッコよくて、クールな GUI に
 - ドロップシャドウ, フェード, アニメーションなど





8位 StAX (Streaming API for XML)

- すぐ使える
- はまれば納得 ◎
- じわじわと効く





8位 StAX

- **どんな機能?**
 - **第3の XML パーサ**
 - プルモデルの XML パーサ
(SAX プッシュモデル DOM オブジェクトモデル)
 - XML ドキュメント作成も
- **どこに使うの?**
 - **大量の XML を効率よく扱う**
 - パースの制御を自分でおこなえる
 - 読み飛ばし, タグの filtering など思いのまま





8位 StAX

- どうやって使うの?
 - タグの一覧表示

```
XMLInputFactory factory = XMLInputFactory.newInstance();
```

```
BufferedInputStream stream = ...;
```

```
XMLStreamReader reader = factory.createXMLStreamReader(stream);
```

```
for (; reader.hasNext(); reader.next()) {  
    int eventType = reader.getEventType();
```

```
    if (eventType == XMLStreamConstants.START_ELEMENT) {  
        System.out.println("Name: " + reader.getName());  
    }  
}
```

```
}
```





7位 java.awt.Desktop

- すぐ使える
- はまれば納得 ◎
- じわじわと効く





7位 java.awt.Desktop

- どんな機能?
 - プラットフォームに依存したアプリの起動
 - ブラウザ, メーラ, エディタ,
 - ファイルの種別ごとに登録されたデフォルトアプリ
- どこに使うの?
 - ネイティブアプリとの連携
 - ファイルマネージャなど
 - デフォルトアプリというところが重要





7位 java.awt.Desktop

- どうやって使うの?
 - ブラウザ, エディタ, デフォルトアプリの起動

```
Desktop desktop = Desktop.getDesktop();
```

```
// ブラウザー起動
```

```
desktop.browse(new URI("http://google.co.jp"));
```

```
// エディター起動
```

```
desktop.edit(new File("DesktopSample.java"));
```

```
// デフォルトのアプリの起動
```

```
desktop.open(new File("sample.swf"));
```





6位 JAX-WS 2.0

- すぐ使える
- はまれば納得 ◎
- じわじわと効く





6位 JAX-WS 2.0

- **どんな機能?**
 - Web サービスのクライアントを作成する
 - 以前は JAX-RPC と呼ばれていたもの
 - Java SE ではサーバ機能はテスト用のものしかない
- **どこに使うの?**
 - Web サービスのクライアント
 - REST も扱えるので、Java で Mash-Up
 - もちろん、SOAP も OK





6位 JAX-WS 2.0

- どうやって使うの?
 - RESTful な Web Services

```
String uri = ...
```

```
QName serviceName = new QName(uri, "...");  
QName portName = new QName(uri, "...");
```

```
Service service = Service.create(serviceName);  
service.addPort(portName, HTTPBinding.HTTP_BINDING, uri);  
Dispatch<Source> dispatch  
    = service.createDispatch(portName,  
                             Source.class,  
                             Service.Mode.PAYLOAD);
```

```
Map<String, Object> reqContext = dispatch.getRequestContext();  
reqContext.put(MessageContext.HTTP_REQUEST_METHOD, "GET");
```

```
Source result = dispatch.invoke(null);
```





11位 - 20位

- 11位 JAXB 2.0
- 12位 Pluggable Annotation Processor
- 13位 和曆
- 14位 System Tray
- 15位 DTrace
- 16位 SwingWorker
- 17位 分割 Verification
- 18位 Splash Screen
- 19位 Locale Sensitive Service API
- 20位 Derby





注目株 DTrace

- どんな機能?
 - Solaris の DTrace が Java もサポート
 - OS も Java も一緒にたにデバッグ
 - GC 開始/終了, メソッド開始/終了 などのプローブ
- どこに使うの?
 - デバッグ, パフォーマンスチューニング
 - Java 単体だけでなく、マシン全体のチューニング
 - 残念ながら Solaris だけ... orz





5位 On Demand Attach

- すぐ使える ◎
- はまれば納得
- じわじわと効く





5位 On Demand Attach

- どんな機能?
 - 起動オプションなしで、いつでも jconsole
 - J2SE 5.0 では起動オプションが必要だった
 - ただし、同一マシン、同一ユーザに限る
- どこに使うの?
 - 保守・運用
 - JMX & MXBean で軽いモニタリング
 - jconsole も機能向上
 - jconsole の拡張も可能





4位 ResourceBundle 機能拡張

- すぐ使える
- はまれば納得 ○
- じわじわと効く ◎





4位 ResourceBundle 機能拡張

- どんな機能?
 - ResourceBundle を制御する
 - ResourceBundle.Control の導入
 - getBundle メソッドの動作を変更させる
- どこに使うの?
 - リソースの柔軟な制御
 - デフォルトの fallback をおこなわない
 - 検索パスの制御
 - XML でリソースファイル記述





4位 ResourceBundle 機能拡張

- どうやって使うの?
 - Default Locale を探さないようにする

```
ResourceBundle bundle =  
    ResourceBundle.getBundle("MyResources",  
        ResourceBundle.Control.getNoFallbackControl(  
            ResourceBundle.Control.FORMAT_DEFAULT));
```

- 探索パスの制御

```
ResourceBundle bundle =  
    ResourceBundle.getBundle("MyResources",  
        new ResourceBundle.Control() {  
            public List<Locale> getCandidateLocales(String base,  
                Locale locale) {  
                return Arrays.asList(locale, locale.ROOT);  
            }  
        });
```





櫻庭が勝手に選んだ特別賞 Attach API

- すぐ使える
- はまれば納得
- じわじわと効く ○





特別賞 Attach API

- どんな機能?
 - 動作中の Java VM に動的にアタッチ
 - Java VM にエージェントを送り込める
 - On Demand Attach も Attach API がベース
- どこに使うの?
 - デバッグ、動的な機能追加など
 - JVMTI のエージェントが使用可
つまりデバッガ、プロファイラが使用できる
 - 後からモジュール追加
 - いろいろ使えそう





3位 OutOfMemoryError

- すぐ使える ◎
- はまれば納得
- じわじわと効く





3位 OutOfMemoryError

- どんな機能?
 - OutOfMemoryError のスタックトレース
 - J2SE 5.0 まではスタックトレースが表示されず
- どこに使うの?
 - デバッグ
 - メモリリークなどの解析にぜひ
 - 設定の必要なし そのまま実行するだけ
 - Java SE 6 以前の JDK でコンパイルされたクラスファイルでも OK





2位 Scripting for Java

- すぐ使える
- はまれば納得
- じわじわと効く ◎





2位 Scripting for Java

- どんな機能?
 - Java から統一的にスクリプトを扱う
 - 標準では JavaScript (Mozilla Rhino) が使用可
 - それ以外は Scripting Project (<https://scripting.dev.java.net>)
- どこに使うの?
 - それが一番難しい
 - Super Glue, Inner Jelly, Outer Jelly Pattern
 - プレゼンテーション層
 - テストケース, 設定ファイル
 - プロトタイプ作成





2位 Scripting for Java

- どうやって使うの?
 - スクリプト実行

```
ScriptEngineManager manager = new ScriptEngineManager();
```

```
ScriptEngine engine = manager.getEngineByName("js");
```

```
// 文字列でスクリプト
```

```
String script = "print('Hello, World!');";  
engine.eval(script);
```

```
// Readerでスクリプトを読みこむ与える
```

```
Reader reader = ...  
engine.eval(reader);
```





2位 Scripting for Java

- どうやって使うの?
 - スクリプトでのオブジェクト作成 (Rhino)

```
// 基本
```

```
var date = new Packages.java.util.Date();
```

```
// 省略形 (java, javax, org ではじまるパッケージ)
```

```
var date = new java.util.Date();
```

```
// クラス (パッケージ) のインポート
```

```
importClass(java.util.Date);
```

```
var date = new Date();
```





2位 Scripting for Java

- どうやって使うの?
 - 既存オブジェクトにアクセス

- Java -

```
Bindings bindings = engine.getBindings(ScriptContext.ENGINE_SCOPE);  
bindings.put("date", new Date());
```

- JavaScript -

```
var millisec = date.getTime();
```





2位 Scripting for Java

- どうやって使うの?
 - インタフェースの実装 (Rhino)

```
importClass(javax.swing.JButton);
var button = new JButton("OK");

var listenerObj = {
    actionPerformed: function(event) {
        print("ActionPerfomed: " + event + "\n");
    }
}

importClass(java.awt.event.ActionListener);
var listener = new ActionListener(listenerObj);

button.addActionLi stener(l i stener);
```





1位 Compiler API

- すぐ使える
- はまれば納得 ○
- じわじわと効く ◎





1位 Compiler API

- どのような機能?
 - プログラム動作中にコンパイル (javac)
 - javac を呼び出せるようになったただけだが...
 - ファイルを使用しなくてもコンパイル可能
- どこに使うの?
 - 実行中のモジュール追加
 - Annotation + Compiler API + ClassLoader
 - すべてオンメモリでも可能
 - ただし, 決して使いやすいわけではない





1位 Compiler API

- どうやって使うの?
 - HelloWorld.java をコンパイル

```
JavaCompiler compiler = ToolProvider.getSystemJavaCompiler();  
StandardJavaFileManager fileManager  
    = compiler.getStandardFileManager(null, null, null);
```

```
Iterable<? extends JavaFileObject> units  
    = fileManager.getJavaFileObjects(new File("HelloWorld.java"));
```

```
JavaCompiler.CompilationTask task  
    = compiler.getTask(null, fileManager, null, null, null, units)  
task.call();
```

```
fileManager.close();
```





Conclusion

- 1位 Compiler API
- 2位 Scripting for Java
- 3位 OutOfMemory
- 4位 ResourceBundle 機能拡張
- 5位 On Demand Attach
- 6位 JAX-WS 2.0
- 7位 java.awt.Desktop
- 8位 StAX
- 9位 JOGL との統合
- 10位 GourpLayout
- 注目株 DTrace
- 特別賞 Attach API





Conclusion

- でも, やっぱり...
 - 自分で使ってみて, 自分なりの Top 10 を
 - 人によって用途は違う 目的も違う
- 新しい機能を使わなくても, SE 6 を!
 - パフォーマンスの向上
 - ソフトウェア管理機能の拡張
 - 移行の痛みは少ないはず

