



Taming the Tiger

The Java™ 2 Platform, Standard Edition, Release 1.5

Mark Reinhold

Senior Staff Engineer

Sun Microsystems, Inc.

mr@sun.com





Agenda

- Quick update on 1.4 maintenance releases
 - 1.4.1 “Hopper”
 - 1.4.2 “Mantis”
- Overview of 1.5 “Tiger”
 - Themes and focus
 - Main features



J2SE Platform Status

- J2SE platform is widely adopted
 - Now long past the “early adopter” phase
- Being used for mission critical apps
 - Often with very demanding scalability needs
- Supports big servers
 - Both web tier and transaction tier
- Supports a wide variety of GUI apps
 - From small applets to very large desktop apps



“Hopper” 1.4.1

- Main focus was on quality improvements
 - Over two thousand bug fixes
- Limited Itanium support
 - Linux and Windows
 - For developers only, no JIT



“Hopper” 1.4.1 *(Continued)*

- New garbage collectors
 - Concurrent mark-and-sweep
 - Separate thread does GC work
 - Parallel young space collection
 - Exploits multiple CPUs
- Deployment improvements
 - Especially for internet applets (games, *etc.*)
 - Reduced download size



“Mantis” 1.4.2

- Again, main focus is on quality
 - Another 2,700 bug fixes!
- Full Itanium support
 - Linux and Windows
 - Fully deployable, high-performance JIT
- Lots of general performance work
 - Especially for Linux/x86
 - Significantly improved startup time (15–30%)



“Mantis” 1.4.2 *(Continued)*

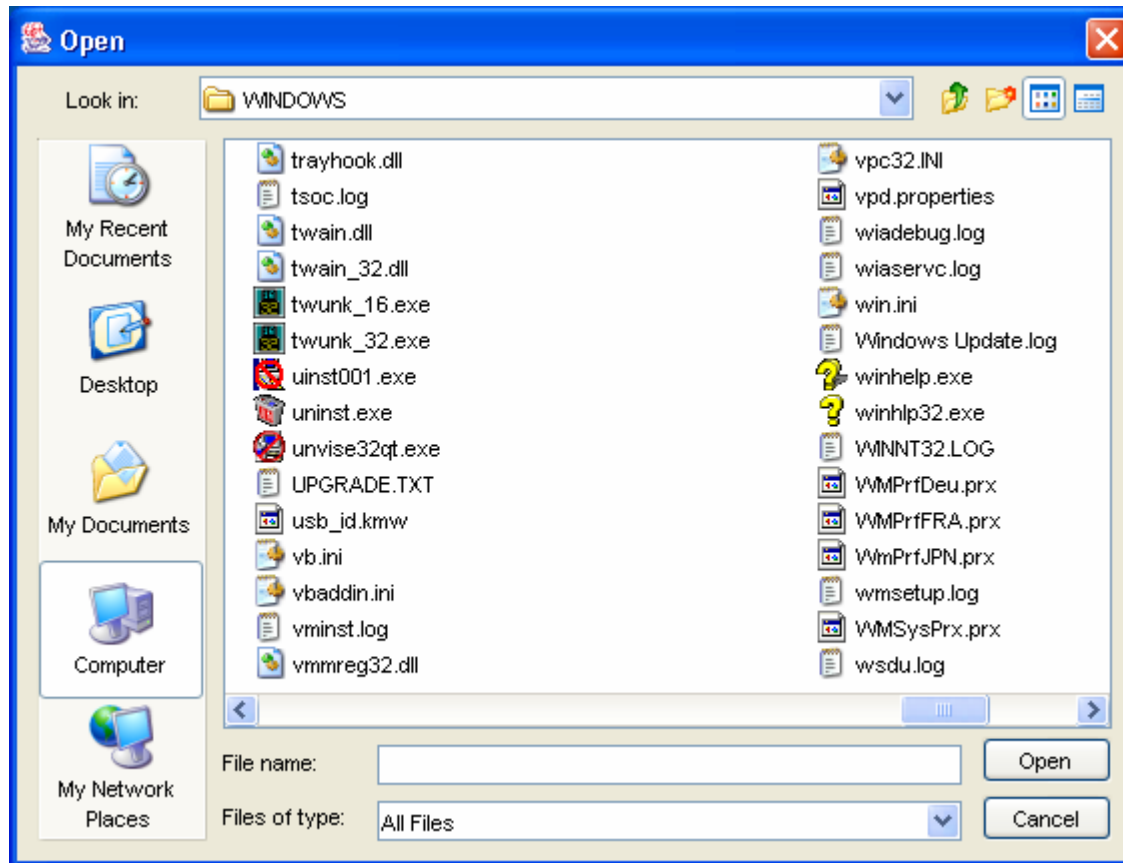
- Two new Swing look-&-feels
 - Windows XP and Linux GTK
- Deployment improvements
 - New windows installer
 - Reduced download size
- The return of the ActiveX bridge



“Mantis” 1.4.2: Deferred

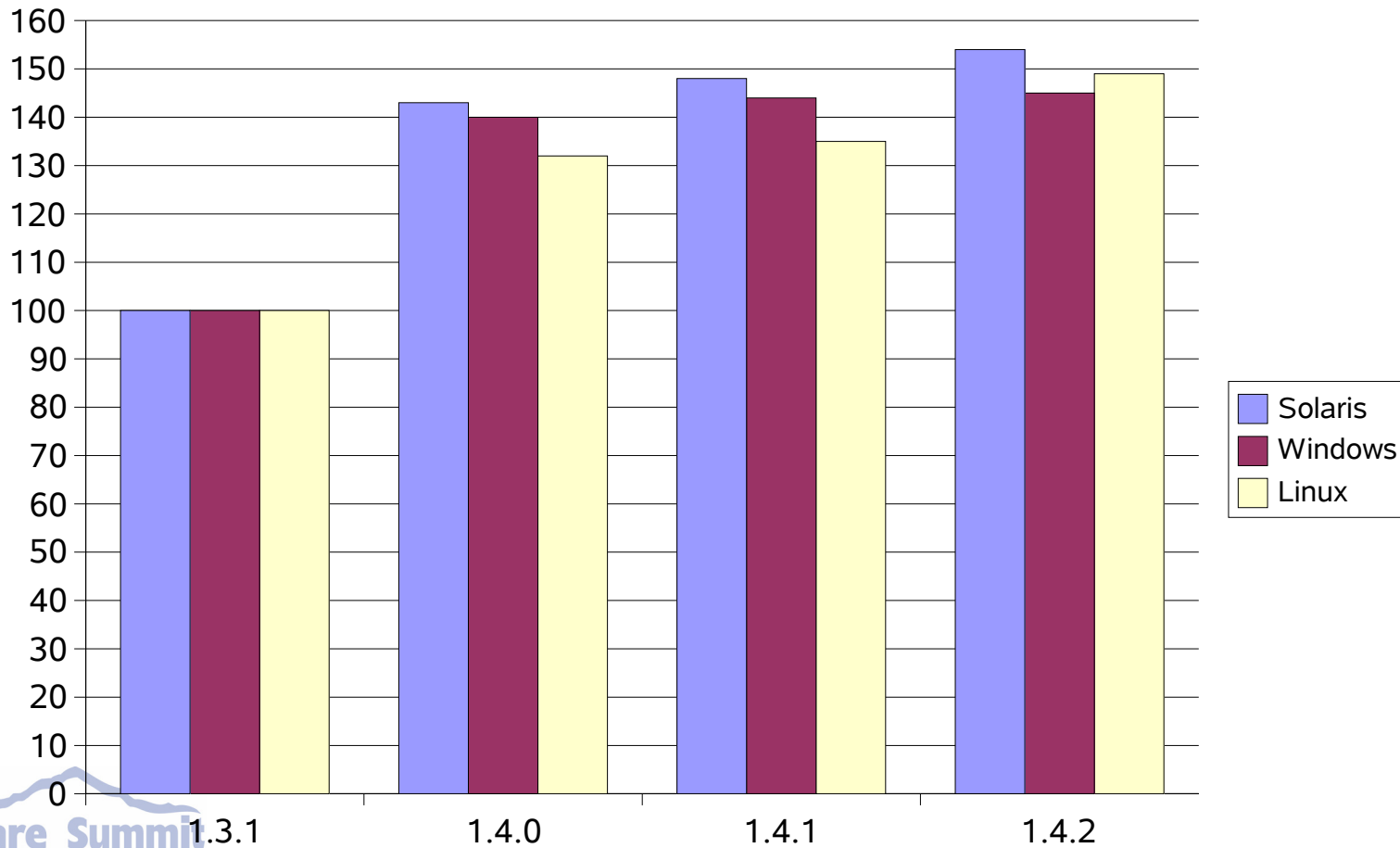
- Auto-update for JRE
 - Targeted at internet consumers
 - Checks for updates periodically
 - Helps users download and install updates
- Was targeted for 1.4.2
 - Needs more test exposure
 - Will deliver in a 1.4.2_**xx** update

Windows XP Look & Feel



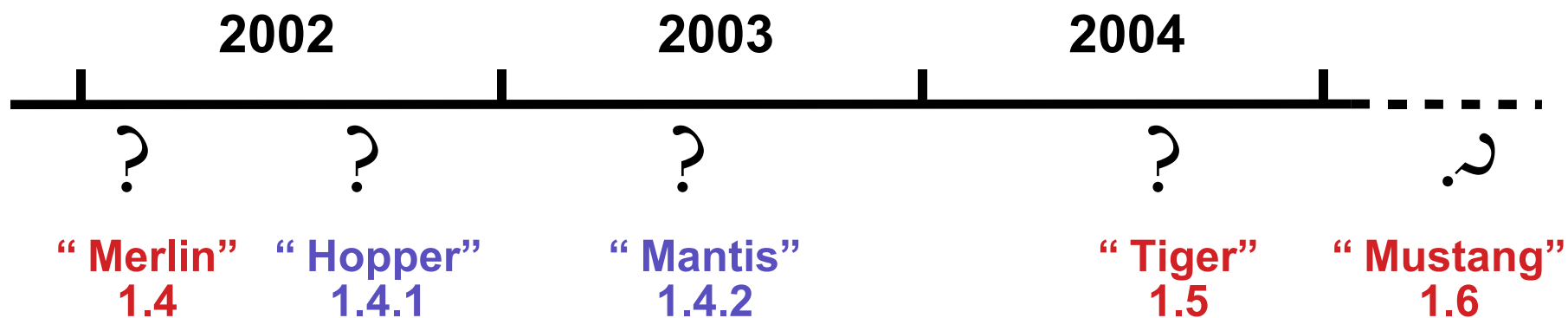
Swing Performance Trends

SwingMark (bigger is better)





Timeline





“Tiger” 1.5

- Next major J2SE release
- Targeted at summer of 2004
- Tiger is **BIG** and **FIERCE!**
 - 15 component JSRs for major new features
 - Many, many upgrades to existing features
 - Lots of bug fixing and performance work
- Specification being developed in the JCP
 - Umbrella JSR 176



Tiger Expert Group

Apache

Apple

Borland

Cisco Systems

Fujitsu Limited

Hewlett Packard

IBM

Macromedia

Nokia

Oracle

SAP AG

SAS Institute

SavaJe

Oswaldo Doederlein

Juergen Kreieder



Tiger Component JSRs

003 JMX Management API

013 Decimal Arithmetic

014 Generic Types

028 SASL

114 JDBC Rowsets

133 New Memory Model

163 Profiling API

166 Concurrency Utilities

174 Monitoring & Mgmt

175 Metadata

199 Compiler API

200 Pack transfer format

201 Minor language updates

204 Unicode surrogates

206 JAXP 1.3



Tiger Balances Many Criteria

- Big focus: Quality, stability, compatibility
 - Customers need rock-solid releases
- Support a wide range of application styles
 - “From desktop to data center”
- Big emphasis on scalability
 - Exploit big heaps, big I/O, big everything
- Continue to deliver great new features
 - But in an evolutionary way



Tiger Themes

- Quality
 - Compatibility, compatibility, compatibility!
- Monitoring and Manageability
- Performance and Scalability
- Core XML Platform
- Ease of Development
- Desktop Client



Quality, Stability, Compatibility

- This talk focuses on new features
 - Because they are easy to talk about
 - But they're really just the icing on the cake
- J2SE team is heavily focused on quality
 - Lots and lots of people working away on bugs
 - We know compatibility is key to customers
 - We know reliability and stability are critical
 - We are making big SQE/SQA investments
- Don't let the feature list distract you from this!



Monitoring & Manageability

- Key component of J2SE RAS strategy (Reliability, Availability, Serviceability)
- JVM Monitoring & Management (JSR 174)
 - Low memory detection
 - Allows access to internal VM status
 - Heap sizes, GC info, threads, *etc.*
 - Industry standard, SNMP based
 - Plugs into existing management consoles

Monitoring & Manageability

(Continued)

- JMX Management API (JSR 003)
 - Support for remote management
 - Works with existing J2EE application servers
- New JVM profiling API (JSR 163)
 - Allows improved performance analysis
- Improved diagnostic facilities
 - Stack trace API, VM error handling



Performance & Scalability

- Even more support for big heaps
 - Improved concurrent/parallel collection
- Yet more HotSpot tuning
 - We want those bytecodes to fly!
- JVM memory sharing to reduce footprint
- Faster startup time
 - Especially important for desktop apps
- Performance ergonomics
 - Less reliance on command-line flags
 - More support for adaptive self-tuning



Core XML Platform

- Tiger will support latest XML standards
 - XML 1.1 and XML Namespace
 - XML Schema
 - XSLT
 - SAX 2.0.1
 - DOM level 3



Web Services Developer Pack

- Core Web Services platform
 - JAX-RPC & SAAJ (WSDL/SOAP)
 - Core infrastructure for Web Services
 - JAXB
 - Easy binding of XML data to Java objects
 - XML Security
 - Encryption and digital signature
 - JAXR
 - API for XML registries
- Not part of Tiger



JDBC RowSets

- **CachedRowSet**
 - Contains in-memory collection of rows
 - Allows disconnected use of data
 - Updates can be resynchronized into database
- **WebRowSet**
 - Uses XML for data transfer

Ease of Development

Metadata

- Key new Java language idea
- Decoration of classes/methods/fields

```
public @Fred @Bert(3) int x;
```
- Supports declarative programming
- It is a new way of thinking
- Will be used to simplify J2EE APIs
 - By defining standard decorations and tools to process them

Ease of Development

Metadata Example: JAX-RPC

```
public interface PingIF extends Remote {  
    public void ping() throws RemoteException;  
}
```

```
public class Ping implements PingIF {  
    public void ping() { ... }  
}
```

Ease of Development

Metadata Example: JAX-RPC

```
public interface PingIF extends Remote {  
    public void ping() throws RemoteException;  
}
```

```
public class Ping implements PingIF {  
    public void ping() { ... }  
}
```

```
public class Ping {  
    public @remote void ping() { ... }  
}
```



Ease of Development

Generic Types

```
// This HashMap maps Strings to Mammals  
HashMap m
```

```
    = new HashMap();
```

```
m.put("wombat", new Mammal());
```

```
Mammal w = (Mammal)m.get("wombat");
```

```
m.put("gecko", new Lizard());
```

```
Mammal x = (Mammal)m.get("gecko");
```

```
★ Runtime error: ClassCastException
```



Ease of Development

Generic Types *(Continued)*

```
// This HashMap maps Strings to Mammals
HashMap<String,Mammal> m
    = new HashMap<String,Mammal>();
m.put("wombat", new Mammal());
Mammal w = m.get("wombat");    // No (Mammal)!
```

```
m.put("gecko", new Lizard());
```

★ **Compile-time error: Type mismatch**



Ease of Development

Other Tiger Language Features

Iterating over collections

```
for (Iterator<String> i = c.iterator();  
     i.hasNext();)  
{  
    String s = i.next();  
    ...  
}
```



Ease of Development

Other Tiger Language Features

Iterating over collections

```
    for (Iterator<String> i = c.iterator() ;  
        i.hasNext() ;)  
    {  
        String s = i.next() ;  
        ...  
    }  
    for (String s : c) { ... }
```



Ease of Development

Other Tiger Language Features

- Iterating over collections

```
for (String s : c) { ... }
```

- Enumerated types

```
public class Toss {  
    private String name;  
    private Toss(String n) { name = n; }  
    public toString() { return name; }  
    public Toss HEADS = new Toss("HEADS");  
    public Toss TAILS = new Toss("TAILS");  
    ...  
}
```

Ease of Development

Other Tiger Language Features

- Iterating over collections

```
for (String s : c) { ... }
```

- Enumerated types

```
public class Toss {  
    private String name;  
    private Toss(String n) { name = n; }  
    public toString() { return name; }  
    public Toss HEADS = new Toss("HEADS");  
    public Toss TAILS = new Toss("TAILS");  
    ...  
}
```

```
public enum Toss { HEADS, TAILS }
```

Ease of Development

Other Tiger Language Features

- Iterating over collections

```
for (String s : c) { ... }
```

- Enumerated types

```
public enum Toss { HEADS, TAILS }
```

- Autoboxing of primitive types

```
Integer x = new Integer(3);  
int y = x.intValue();  
map.put(new Integer(1),  
        new Integer(42));
```

Ease of Development

Other Tiger Language Features

- Iterating over collections

```
for (String s : c) { ... }
```

- Enumerated types

```
public enum Toss { HEADS, TAILS }
```

- Autoboxing of primitive types

```
Integer x = new Integer(3);  
int y = x.intValue();  
map.put(new Integer(1),  
        new Integer(42));
```

```
★ Integer x = 3;  
★ int y = x;  
★ map.put(1, 42);
```

Ease of Development

Simple Formatting & Scanning

- `printf/scanf` popular in C/Perl/*etc.*
 - Powerful, easy to use
- Adding similar features to J2SE 1.5

```
out.format("%-12s is %2d long", name, l);
out.format("value = %6.3f", value);
out.format("What is your favorite number? ");
int n = in.nextInt();
```
- Fully type-safe at run-time



Desktop Client

- Additional accessibility support
- Continue to improve deployment
 - Upgrades to Java Plug-in and Java WebStart
- Continued quality & performance work
 - Startup and footprint work
 - Direct support for OpenGL
- Improved GUI look-and-feel support
 - Gnome skins support
 - Better Windows look and feel
 - Revised Java look and feel



Unicode Supplementary Chars

- Unicode 3.1 adds more characters
 - Some characters don't fit in 16 bits
 - *Yikes!*
- Java `char` will stay at 16 bits
- Extended chars will be pair of values
 - Encoded in strings
- Library APIs will support these chars
 - As pairs within strings



Miscellany

- Concurrency Utility Library (JSR 166)
 - Thread pools, semaphores, ...
 - Being led by Doug Lea



Taming the Tiger

*The Java™ 2 Platform,
Standard Edition,
Release 1.5*

- And lots more...
 - There's way too much for any one talk!



Taming the Tiger

*The Java™ 2 Platform,
Standard Edition,
Release 1.5*

Call to action

- Check out the JCP reviews still in progress
 - <http://jcp.org/jsr/detail/176.jsp>
- Download the beta release when it ships
 - <http://java.sun.com/j2se/1.5>
 - Does your old stuff work okay?
 - Do the new APIs meet your needs?
- Tell us what you think!
 - <http://java.sun.com/cgi-bin/bugreport.cgi>
 - [http://java.sun.com/mail/](http://java.sun.com/mail/j2se-comments@java.sun.com)
 - j2se-comments@java.sun.com