



# Dateizugriff mit new I/O 2

Wolfgang Nast

Nürnberg, 21.11.2012

# MT AG

## Daten und Fakten

Als anerkannter Partner der führenden Technologiehersteller verbinden wir die Agilität eines mittelständischen Unternehmens mit der Lösungskompetenz internationaler Beratungshäuser.

### Hauptsitz

Ratingen

### Niederlassungen

Hamburg, Dortmund, Frankfurt,  
Luxemburg

### Tochtergesellschaften

MT-ifs GmbH, MT-ics GmbH

### Beschäftigte (2012)

220 Festangestellte  
80 Freiberufler



# Dateizugriff mit new I/O 2

- Neue Elemente
- Benutzer und Rechte
- Hard- und Softlinks
- Suchen im Dateisystem
- Überwachen des Ordners

# Neue Elemente

- **Holen des Standard-Dateisystems:**

```
FileSystem fs = FileSystems.getDefault();
```

- **Holen der "Root"-Directories:**

```
Iterable<Path> pList = fs.getRootDirectories();
```

- **Datei „home\unterordner\Datei.txt“ ansprechen:**

```
Path p1 = fs.getPath("home", "unterordner", "Datei.txt");
```

- **Verzeichnisse/Links im „Path“ ansprechen:**

```
Iterator<Path> i = p1.iterator();
```

- **Softlink auflösen zu realem Verzeichnis:**

```
Path p2 = p1.toRealPath();
```



# Berechtigungs-System

# Berechtigung NTFS(ACL)

- **Besitzer ermitteln:**

```
AclFileAttributeView acl = Files.getFileAttributeView(p2,  
AclFileAttributeView.class);  
    System.out.println("Besitzer = " +  
acl.getOwner().getName());
```

- **Benutzer suchen:**

```
UserPrincipal userP =  
p1.getFileSystem().getUserPrincipalLookupService().  
lookupPrincipalByName("Benutzer");
```

- **Gruppe suchen:**

```
GroupPrincipal grP =  
p1.getFileSystem().getUserPrincipalLookupService().  
.lookupPrincipalByGroupName("admin");
```

# Rechte lesen und erstellen

## NTFS (ACL)

- Lesen der Rechte

```
for(AclEntry aclEntry : acl.getAcl()){  
    System.out.println("rechte: " + aclEntry.toString());  
}
```

- Recht zusammenbauen:

```
AclEntry aclEnt1 = AclEntry.newBuilder()  
    .setType(AclEntryType.ALLOW)  
    .setPrincipal(userP)  
    .setPermissions(AclEntryPermission.READ_DATA,  
AclEntryPermission.WRITE_DATA)  
    .build();
```

# Rechte setzen

## NTFS (ACL)

- Bei ALLOW am Anfang, vor allen DENY-Einträgen:

```
acl.getAcl().add(0, aclEnt1);
```

- Bei DENY am Ende, nach allen ALLOW-Einträgen:

```
acl.getAcl().add(aclEnt1);
```



# Berechtigung UNIX (POSIX)

- **Besitzer ermitteln:**

```
PosixFileAttributes at = Files.readAttributes(file,
PosixFileAttributes.class);
System.out.println("Besitzer = " +
at.getOwner().getName());
```

- **Rechte setzen:**

```
at.setPermissions(PosixFilePermissions.fromString("rw-r--r--"));
```

- **Oder:**

```
Set<PosixFilePermission> permSet = new HashSet<>();
permSet.add(PosixFilePermission.OWNER_READ);
permSet.add(PosixFilePermission.GROUP_EXECUTE);
...//weitere Rechte
at.setPermissions(permSet);
```



# Neue Verzeichnis Tools

# Visitor Interface

- `FileVisitResult postVisitDirectory(T dir, IOException exc)`
- `FileVisitResult preVisitDirectory(T dir, BasicFileAttributes attrs)`
- `FileVisitResult visitFile(T file, BasicFileAttributes attrs)`
- `FileVisitResult visitFileFailed(T file, IOException exc)`

- **Einfacher Aufruf:**

```
Files.walkFileTree(p1, vis);
```

- **Umfangreicher Aufruf:**

```
int tiefe = 20;  
EnumSet<FileVisitOption> links =  
EnumSet.of(FileVisitOption.FOLLOW_LINKS);  
Files.walkFileTree(p1, links, tiefe, vis);
```

# PathMatcher

- Vergleichen mit „regex“ oder „glob“

```
PathMatcher mat = p1.getFileSystem().getPathMatcher(  
    "glob: *.{java,class}");  
if (mat.matches(p2)) {  
  
    ...  
  
    //Weitere Aktionen  
  
    ...  
  
}
```

# Verzeichnis-Überwachung

- EventTypen:

```
OVERFLOW, ENTRY_CREATE, ENTRY_DELETE, ENTRY_MODIFY
```

- WatcherService registrieren:

```
WatchService watcher =  
p1.getFileSystem().newWatchService(); p1.register(watcher,  
StandardWatchEventKinds.ENTRY_DELETE);
```

- Auf Ereignis warten:

```
key = watcher.take();
```

- Ereignis auswerten:

```
for (WatchEvent<?> event : key.pollEvents()) {  
...  
boolean valid = key.reset();
```



# Nützliches

# Hard- und Softlinks anlegen

- Softlink anlegen:

```
Files.createSymbolicLink(neuerLink, linkZiel);
```

- Hardlink anlegen:

```
Files.createLink(neuerLink, linkZiel);
```

# Verzeichnisse kopieren und verschieben

- Verzeichnis kopieren:

```
Files.copy(original, kopi, REPLACE_EXISTING,  
COPY_ATTRIBUTES, NOFOLLOW_LINKS);
```

- Verzeichnis verschieben:

```
Files.move(ursprung, ziel, REPLACE_EXISTING, ATOMIC_MOVE);
```





Fragen ?

# Besuchen Sie auch unsere weiteren Vorträge auf der DOAG 2012

Dienstag, 12 Uhr, Raum Riga

Dienstag, 13 Uhr, Raum Seoul

Dienstag, 14 Uhr, Raum Stockholm

Dienstag, 15 Uhr, Raum Kopenhagen

Dienstag, 16 Uhr, Raum Stockholm

Mittwoch, 13 Uhr, Raum Riga

Mittwoch, 15 Uhr, Raum Riga

Mittwoch, 16 Uhr, Raum Seoul

Donnerstag, 09 Uhr, Raum Istanbul

Donnerstag, 14 Uhr, Raum Konf. EG

Donnerstag, 15 Uhr, Raum Istanbul

Donnerstag, 16 Uhr, Raum Oslo

Dynamisch Unterschiede in Datensätzen auf Feldebene finden by S.O. Kelbert

Route to ASM by Ernst Leber

Automatische Generierung der ETL-Prozesse OWB vs. ODI by Irina Gotlibovych

Wiederverwendung von bestehendem PL/SQL Code in ADF Anwendungen by Hendrik Gossens

„Managed Code“ mit OWB – Methoden und Wege by Bernhard Rosenberger

Dateizugriff mit new I/O 2 by Wolfgang Nast

WebServices in Java SE und EE by Wolfgang Nast

Das Mysterium OPatch by Volker Mach

Das größte APEX Projekt der Welt @ Union Investment by Niels de Bruijn

Testen mit Pfefferminzgeschmack by Birgit Kratz

APEX goes UNIT Testing by Oliver Lemm

SOA verspielt – rekursive BPEL Prozesse by Guido Neander



# Vielen Dank.

**MT AG**

Balcke-Dürr-Allee 9  
40882 Ratingen

Telefon: +49 (0) 21 02 309 61-0  
Telefax: +49 (0) 21 02 309 61-10

E-Mail: [info@mt-ag.com](mailto:info@mt-ag.com)  
[www.mt-ag.com](http://www.mt-ag.com)