

Kunden warten manchmal viele Jahre, bevor sie ein Migrationsthema anfassen. Das ist besonders in den Fällen verständlich, in denen von erprobter und stabiler Client/Server-Technologie in die neue komplexe Welt der Drei-Schichten-Architektur gewechselt werden soll. Wenn eine komplexe Software mit vielen Masken, Berichten und Schnittstellen zu migrieren und ein reibungsloser Wechsel in einer neue Welt gewünscht sind, müssen alle Details vorher genau untersucht werden. Dieser Artikel gibt einige Erfahrungen weiter und dem Entwickler ein paar hilfreiche Tipps an die Hand.

# Erfahrungen bei der Migration von Forms/Reports 6i nach Forms/Reports 11.1.2 und 12

Frank Christian Hoffmann, Cologne Data GmbH

Ende Oktober 2011 ist auf den bekannten Download-Seiten von Oracle die hier getestete Version von Forms und Reports unter der Gruppe „Fusion Middleware“ aufgetaucht. Dort befinden sich auf weiteren Reitern Dokumentationen und Demos. Ein guter Einstieg für jeden, der sich in die Materie einlesen will, findet sich unter den folgenden Links:

- <http://www.oracle.com/technetwork/developer-tools/forms/overview/index.html>
- <http://www.oracle.com/technetwork/middleware/reports/overview/index.html>

## Neuerungen im Release 11g R2

Die neue Version 11.1.2 verspricht neben zahlreichen Bugfixes eine Aufteilung von Entwickler- und Server-Installation, eine Client-Entwicklerunterstützung von Windows XP (32-Bit) bis hin zu Windows 7 (64-Bit-Systeme), Verbesserungen in der Monitoring-Software, Verbesserungen bei der Anzeige von „NCHAR“-Spalten sowie zahlreiche Fehler-Korrekturen. Eine sehr gute Optimierung stellt die Speichernutzung der Entwicklerumgebung dar. Die neue Umgebung 11.1.2 kommt mit etwa 800 MB Hauptspeicherbedarf für WebLogic sowie für Forms und Reports aus. Das ist zu den gut 3,5 GB der Vorgängerversion ein Schritt in die richtige Richtung.

Einen ersten Überblick für das Migrationsvorhaben liefert der knapp 60 Seiten umfassende Upgrade-Guide von Forms 6i nach Forms 11g (E24487-01). Hier kann man sich einen guten Überblick über die Anforderungen der Migration verschaffen. Wer es etwas genauer wissen will, kann dann die Deployment-Guides für Forms und Reports lesen und weitere knapp tausend Seiten studieren. Alle Dokumente lassen sich unter den beiden angegebenen Web-Links finden. Übersichtlich und praktisch erscheint der Download als PDF-File. Für ein erstes „Look&Feel“ sind auch die Demos und die Dokumentation dazu hilfreich.

Eine Migration kann aus Sicht des Autors durchgeführt werden, wenn alle kritischen Punkte des Software-Updates zuvor geklärt und Lösungen dazu gefunden wurden. Das kann dazu führen, dass Teile der Software neu geschrieben sowie Java-Funktionen ergänzt werden müssen und man sich von liebgewonnenen Benutzerdialogen verabschieden muss.

## Schritt 1: Analyse der bisherigen 6i-Forms/Report-Software

Der Upgrade-Guide hilft, die Knackpunkte der Migration zu ermitteln. Überall dort, wo Schnittstellen zu anderen Software-Produkten wie Text- oder Tabellenverarbeitung, C-Programmen, Browser-Aufrufen und Erstellung oder Import von Dateien bestehen, wird eine

Anpassung notwendig sein. Eine Text-Suche mit einem geeigneten Tool nach Schlüsselbegriffen hilft hier weiter.

Wenn viele Änderungen notwendig sind, stellt sich die Frage nach der automatisierten Anpassung. Das kann eine externe Firma durchführen oder man macht sich selbst mit Hilfstools auf den Weg. Hinweise zu den Tools und Firmen, die eine Analyse unterstützen, hat Oracle unter „Customers und Partners“ auf der Forms-Seite zusammengestellt (siehe <http://www.oracle.com/technetwork/developer-tools/forms/learnmore/index.html>).

Soll die zukünftige Report/Forms-Welt unter Unix laufen, muss die komplette Software auf Groß- und Kleinschrift geprüft werden (Icons, Objekt- und Bibliotheks-Referenzen). Dafür empfiehlt der Autor den „Forms-API-Master“, der mit einem Skript diese komplette Arbeit in wenigen Minuten erledigt. Zudem muss man beachten, dass die Forms-Sessions dann in einer Unix-Umgebung laufen – alle „host“-Kommandos sind dann sicherlich auch zu prüfen und eventuell anzupassen.

## Schritt 2: Auswahl von Software-Lösungen in der neuen Drei-Schicht-Umgebung

Es entstehen unterschiedliche Lösungsszenarien, abhängig davon, ob ein Unix- oder ein Windows-Server als Applikations-Server zum Einsatz kommt,

ob man seine Reports im Browser oder direkt als PDF-File mit einem PDF-Reader angezeigt bekommen will und welche Applet-Technologie zum Einsatz kommen soll. Es sind viele weitere Details zu beachten, wenn man eine nicht Oracle-zertifizierte, browserunabhängige Lösung mit „Java Web Start“ oder eine Barrierefreiheit über die Java-Bridge erreichen will. Hier empfiehlt der Autor den Lösungsansatz von Tom Kyte, für jede Anforderung ein kleines Beispiel zu entwickeln. Diese Beispiele sollten dann mit Anwendern und Entwicklern diskutiert und abgestimmt werden, damit alle in dem Migrationsprozess mitgenommen werden. Das hält dann etwas auf, dient aber auf jeden Fall der Akzeptanz des Vorhabens.

### Schritt 3: Konfiguration von Server, Entwickler und Anwenderarbeitsplätzen

Das klingt nur nach etwas Software-Installation – ist jedoch in der Praxis sehr aufwändig. Wichtig ist, dass dieser Bereich nicht unterschätzt wird. Das Ganze fängt damit an, dass man die Zertifizierungsmatrix studiert und sich ein System überlegt, das auch von Oracle zertifiziert ist. Dann plant man eine zusätzliche Hardware-Beschaffung für den Applikations-Server und spart schon mal sein Geld für die dafür notwendigen Oracle-Lizenzen. Aber auch technische Fragen müssen zuvor beantwortet sein: „Wie viele Forms-Sessions müssen auf meinen Applikations-Server?“, „Deckt ein Server das alles ab?“ oder „Wie ist meine Backup- und Recovery-Strategie für meinen Applikations-Server?“ Hinzu kommen Software-technische Fragen: „Wo lege ich meine neuen Applikations-Parameter für Entwickler-, Test- und Produktions-Umgebung ab (Server-Namen, Report-Server-Namen, Verzeichnisse etc.)?“, „Wie führe ich meine Applikations-Software-Updates durch?“, „Wie kann ich prüfen, wer angemeldet ist, und notfalls Forms-Sessions stoppen, ohne den Applikations-Server anhalten zu müssen?“, „Wie verteile ich JAR-Files (etwa für Forms-eigene oder Windows-Komponenten)?“, „Wie richte ich einen Entwickler-Arbeitsplatz ein?“, „Wie soll ein Anwender-Arbeitsplatz

aussehen?“, „Wo generiere ich Dateien aus einer Forms-Session?“, „Wie lade ich Dateien auf den Client und in welche Verzeichnisse?“, „Welches Konzept nutze ich beim Report-Server?“, „Wie werden Reports auf den Client geladen, wo liegen sie und wie sicher ist das neue Konzept?“, „Wie laufen zukünftig Massen-Drucke?“, „Wie signiere ich meine JAR-Files, damit sie geladen werden können?“, „Wenn Entwickler unter Windows entwickeln und die Software unter Unix verteilt werden soll – wie bilde ich den passenden Workflow dazu ab?“ oder „Wie richte ich Bildschirmauflösung, Farbschema, Hintergrundfarben, Logos etc. ein?“ Es ist ein wirklich komplexes Feld, das viel Zeit in Anspruch nimmt.

### Schritt 4: Durchführung der Migration

Wenn die ersten drei Schritte erfolgreich absolviert wurden, sollte der vierte Schritt nur noch eine Ausführung sein. Ein ordentliches Konzept wird dann auch eine vernünftige Aufwandsschätzung für die Migrationsarbeiten beinhalten und könnte auch als Feinkonzept und Styleguide für eine Ausschreibung herhalten. So wird auf jeden Fall am Ende ein Ergebnis erreicht, das den gewünschten Vorstellungen entspricht und abgenommen werden kann.

Für den Entwickler empfiehlt sich auf jeden Fall, einen Account für Oracle Support zu bekommen. Hier findet man aktuelle Java-Files („JAR“) sowie die Hinweise zur Signierung und zu möglichen Problemstellungen. Weiterhin empfiehlt sich das Oracle-Forms-Forum für Fragen und Austausch rund um das Thema. Zu empfehlen ist auch das deutsche Oracle-Forms-Forum unter <http://www.oracle.com/webfolder/technetwork/de/community/forms/index.htm>. Der Code für die nachfolgenden Tipps steht auf der Webseite [www.cologne-data.de/oracle.html](http://www.cologne-data.de/oracle.html). Weitere Tipps von den Lesern nimmt der Autor gerne in diese Liste auf.

#### Tipp 1: Installation eines Entwicklungsrechners unter Windows 7, SP1 (64x)

Es sind folgende Schritte erforderlich:

1. Forms/Reports 6i (falls man die alte Version noch weiter testen will) plus den Reports-Patch aus Patch 3 einspielen (nnb60.dll in „bin“ kopieren) sowie WinHlp32.exe für Windows 7 (64-Bit-Version) von Microsoft in Deutsch (damit man die Hilfefunktion weiter nutzen kann)
2. Java SE SDK 6 in Version 29 (64 Bit) unter „c:\java\sdk64“ sowie während der Installation JRE (64 Bit) unter „c:\java\jre64“ installieren
3. WebLogic 10.3.5 unter dem installierten 64-Bit-Java-„bin“-Verzeichnis mit „java -jar wls1035\_generic.jar“ einrichten
4. Development-Version Forms/Reports 11g R2 installieren
5. Den passenden Datenbank-Client (32 Bit) installieren
6. Installation der diversen 3rd-Party-SQL-Client-Tools
7. Installation diverser Browser zum Testen (IE9 32 Bit und 64 Bit, Firefox 8, Chrome)
8. Installation Java Realtime Environment (JRE) Version 29 (32 Bit) unter „c:\java\jre32“

#### Tipp 2: Timeout.jar

Dieses Bean von Frank Nimphius aus den Demos löst die Forms-6i-Timer-Events ab, die viel zu viel Netzwerk-Traffic verursachen. Das JAR-File und den Code gibt es bei Oracle Support.

#### Tipp 3: Anpassung von Icons

Oracle liefert einen kleinen Satz von GIF-Icons mit dem JAR-File „*frmall.jar*“ aus. Viele Applikationen haben jedoch eigene Icons für Toolbar und Masken. Diese müssen in GIF- oder JPG-Dateien konvertiert werden. GIF-Files, die in der MDI-Toolbar verwendet werden, müssen „links oben“ und GIF-Files, die in der Maske stehen, „mittig zentriert“ ausgerichtet sein. Am besten nimmt man ein gutes Grafikprogramm und führt diese Arbeit bei transparentem Hintergrund durch.

Werden Icons sowohl in der Toolbar als auch in der Maske verwendet, sind zwei Versionen zu erzeugen. Der Autor gibt daher den in seiner Applikation verwendeten, eigenen „MDI-

Icons“ das Präfix „mdi“. Das zweite Problem besteht darin, dass diese Grafiken nicht so einfach dargestellt werden. Entweder sind sie in einem dem WebLogic-Server bekannten Pfad abgelegt und in der Datei „Registry.dat“ (beispielsweise „webutil/icons“) zugewiesen (siehe Listing 1) oder besser in einem signierten JAR-File, damit das Java-Applet lokal darauf zugreifen kann (siehe Listing 2).

```
registry.dat default.icons.  
iconpath=webutil/icons  
default.icons.iconextension=gif  
formsweb.cfg codebase=
```

Listing 1: Variante A (nur zum Testen)

```
jar -cvf meine_icons.jar *.gif  
formsweb.cfg  
codebase=imagebase
```

Listing 2: Variante B (Produktion)

#### Tipp 4: Eine Java-Applet-Einstellung für alle Browser

Jeder, der mit verschiedenen Browsern experimentiert, ärgert sich über die Standard-Einstellungen in der Konfigurationsdatei „formsweb.cfg“ nach der Installation der Software. Neuere Java-Versionen werden zum Teil ignoriert, einige Browser kennen das Java-Applet nicht und bei manchen Browsern startet die Anwendung noch nicht einmal. Listing 3 zeigt eine Einstellung, die ausgehend von der JRE-Version 1.6.0.29, für alle gängigen Browser funktioniert. Am besten installiert man zuerst alle Browser, die man testen will, und spielt unter „http://java.com/de/download/index.jsp“ dann die aktuelle Java-JRE-Version auf. Die Parameter „jpi\_classid“ und „jpi\_codebase“ sind nur für Microsoft relevant. Die Mimetype-Einstellungen sind für Firefox und Microsoft unterschiedlich und mit einem Semikolon getrennt, aber für alle Java-1.6x-Versionen möglich.

#### Tipp 5: Kleines Batch-File zur Kompilierung aller Sourcen

Alle Oracle-Forms-Module sind nur Upgrade-fähig, das heißt, wenn ein Source-

```
jpi_classid=clsid:CAFEEFAC-0016-0000-FFFF-ABCDEFEDCBA  
jpi_codebase=http://java.sun.com/update/1.6.0/jinstall-6-windows-i586.cab#Version=1,6,0,29  
jpi_mimetype=application/x-java-applet;version=1.6
```

Listing 3: Windows-Einstellungen in „formsweb.cfg“

```
Echo compiling Reports..  
for %%f in (*.rdf) do rwconverter userid=user/passwort@tns  
stype=RDFFILE source=%%f dtype=REPFILE dest=%%f logfile=compilation_  
log.log overwrite=YES batch=YES  
  
Echo compiling Libraries  
for %%f IN (*.pll) do frmcmp userid= user/passwort@tns module=%%f  
batch=yes module_type=library compile_all=yes window_state=minimize  
  
Echo compiling Menu  
for %%f IN (*.mmb) do frmcmp userid= user/passwort@tns module=%%f  
batch=yes module_type=menu compile_all=yes window_state=minimize  
  
Echo compiling Forms....  
for %%f IN (*.fmb) do frmcmp userid= user/passwort@tns module=%%f  
batch=yes module_type=form compile_all=yes window_state=minimize
```

Listing 4: Das Batch-File

```
120 : 0 : „F9“ : 29 : „Werteliste“  
81 : 2 : „Ctrl+Q“ : 32 : „Beenden“  
112 : 2 : „Ctrl+F1“ : 35 : „Tastenbelegung“  
121 : 0 : „F10“ : 36 : „Speichern“  
114 : 1 : „Shift+F3“ : 61 : „Nächster Primärschlüssel“  
115 : 1 : „Shift+F4“ : 62 : „Datensatz entfernen“  
117 : 1 : „Shift+F6“ : 63 : „Datensatz löschen“  
115 : 0 : „F4“ : 64 : „Datensatz duplizieren“  
117 : 0 : „F6“ : 65 : „Datensatz einfügen“
```

Listing 5: Auszug aus der Datei „frmweb.res“

code einmal unter 11g R1 oder R2 kompiliert und gespeichert wurde, kann er nicht mehr für eine Vorgängerversion wiederverwendet werden. Daher sollte man nicht die Quelle migrieren, sondern immer nur eine Kopie davon, um die alten Sourcen zu erhalten. Listing 4 ist ein kleines Batch-File zur Konvertierung aller Sourcen. Es muss im Sourcen-Verzeichnis gestartet werden. Zuvor ist natürlich ein „tnsnames“-Eintrag für die Datenbank unter Forms 11g im Oracle „CONFIG“-Verzeichnis erfolgt, am besten für die Datenbank 11g in einer Zeile ohne Returns und Leerzeichen:

```
tns = (DESCRIPTION = (ADDRESS_  
LIST = (ADDRESS=(PROTOCOL=TCP)  
(HOST=129.247.1.1)(PORT=1526)))  
(CONNECT_DATA=(SERVICE_NAME =  
oraent)))
```

#### Tipp 6: Tastatur-Umsetzung in Web-Forms

Um das gleiche Tastaturverhalten wie in Forms 6i zu erreichen, muss ein Terminal-File zugewiesen werden, das die Tastatur-Eingaben richtig umsetzt. Die Zuordnung erfolgt in formsweb.cfg: „term=frmweb.res“. Listing 5 zeigt einen Auszug aus der Datei „frmweb.res“ im Forms-Verzeichnis.

### **Tipp 7: Das neue Report-Plug-in „Blob-Destination“ (seit 11g R1)**

Der Report-Server kann einen Report seit Version 11g R1 auch direkt in eine Tabelle mit einer BLOB-Spalte (Binary Large Object) schreiben. Damit lässt sich der Report-Prozess besser steuern, man kann den Reports eigene IDs und Namen geben und sie danach browser-unabhängig direkt starten. Das Plug-in muss registriert sein, ein aktuelles 11g-JAR-File von Oracle Support heruntergeladen werden und die Aufrufe brauchen ein paar zusätzliche Parameter.

### **Tipp 8: Applet-Start browserunabhängig – Java Web-Start**

Dieser Tipp ist nicht von Oracle zertifiziert und sollte deshalb zuvor durch eingehende Tests geprüft werden. Folgende Datei (siehe Listing 6) ist als „.jnlp“ zu speichern. Web-Start ist bereits in der Java-Realtime-Umgebung von 1.6 ab Version 12 integriert. Die Lösung ist denkbar, wenn Tipp 7 umgesetzt wird und Reports nicht mehr über „web.show\_document“ in Browsern dargestellt werden. Der Autor hat das bisher nur mit einer Maske getestet und kann noch nicht viel dazu sagen.

### **Tipp 9: Forms-Builder 11.1.2 im „Look & Feel“ von 6i darstellen**

Wenn man in der Registry „HKEY\_LOCAL\_MACHINE/SOFTWARE/Oracle“ von Forms 11.1.2 die nachfolgende Zeile einfügt, wird aus der blauen Forms-Builder-Darstellung wieder eine wie in Forms 6i: „FORMS\_CLAF = TRUE“.

### **Tipp 10: Ein Java-Applet-Fenster aus dem Browser starten (Barrierefreiheit)**

Wer das Applikationsfenster nicht in einem Browser, sondern als Extra-Applet darstellen möchte, wählt im „webforms.cfg“ die Anweisung „separateFrame=True“. Damit wird Forms auch barrierefrei, weil ein Screenreader über die Java-Bridge nun Titel, Spalten und Felder vorlesen kann. Mehr dazu auf den Oracle-Seiten unter „Accessibility“. Nachteil dabei bleibt, dass das Browserfenster auch beim

```
<jnlp spec="1.0+" codebase="http://host:7001/forms/java">
<information>
<title>Meine Applikation</title>
<vendor>Meine Firma</vendor>
<description>Forms Application - Java Web Start</description>
<homepage href="http://host:7001" />
</information>
<security>
<all-permissions />
</security>
<resources>
<j2se version="1.6+" />
<jar href="frmall.jar" main="true" />
</resources>
<applet-desc main-class="oracle.forms.engine.Main" name="Forms"
width="1350" height="1010">
<param name="separateFrame" value="false" />
<param name="lookAndFeel" value="oracle" />
<param name="splashScreen" value="false" />
<param name="colorScheme" value="swan" />
<param name="serverURL" value="http://host:7001/forms/
lservlet?ifcfs=http://VPCE001:7001/forms/frmservlet" />
<param name="serverApp" value="default" />
<param name="logo" value="false" />
<param name="envFile" value="default.env"/>
<param name="baseHTML" value="webutilbase.htm"/>
<param name="baseHTMLjpi" value="webutiljpi.htm"/>
<param name="WebUtilArchive" value="frmwebutil.jar,jacob.
jar,timeout.jar"/>
<param name="WebUtilLogging" value="off"/>
<param name="WebUtilLoggingDetail" value="normal"/>
<param name="WebUtilErrorMode" value="Alert"/>
<param name="WebUtilDispatchMonitorInterval" value="5"/>
<param name="WebUtilTrustInternal" value="true"/>
<param name="WebUtilMaxTransferSize" value="16384"/>
<param name="serverArgs" value="module=start.fmx userid=user/pwd@tns
debug=no buffer_records=no debug_messages=no array=no query_only=no
render=no host= port= record= tracegroup= log= usesdi=yes
param_jnlp=TRUE" />
</applet-desc>
</jnlp>
```

Listing 6: Die „.jnlp“-Datei

Schließen von Forms offen bleibt und die Report-Darstellung von Browser zu Browser sehr unterschiedlich erfolgt. Einzige Ausnahme bildet der Firefox-Browser. Dort kann man in der Version 8.0 das PDF-Plug-in „Native“ einstellen. Der Aufruf eines PDF-Files aus dem Browser startet direkt den PDF-Reader.

### **Tipp 11: Dateien auf dem Client mit „RUN32.DLL“ starten**

Dokumente mit Endungen wie „.docx“ oder „.pdf“ können auch mit auf lokaler Betriebssystemebene verknüpften Programmen gestartet werden. Folgender Aufruf erlaubt die Anzeige eines ge-

nerierten Dokuments. Dieser setzt die Installation von „WEBUTIL“ voraus und erwartet das Dokument in dem Standardbenutzer-Verzeichnis des angemeldeten Anwenders:

- webutil\_host.NonBlocking („rundll32 url.dll, FileProtocolHandler, ||
- webutil\_clientinfo.get\_system\_property(user.home') || ,\' || p\_filename);

Zuvor wurde das Dokument mit „webutil“ angelegt: „... client\_text\_io.fopen(webutil\_clientinfo.get\_system\_property(user.home') || ,\' || p\_filename,'W');“

**Tipp 12: Anpassung von HOST-Befehlen**

Windows7 speichert standardmäßig in der 64-Bit-Version 32-Bit-Software im Verzeichnis „Program Files (x86)“ und 64-Bit-Software unter „Program Files“. Die Sonderzeichen können beim Aufruf alter CMD-Befehle zu Fehlern führen. Dieser Tipp umgeht diese Einschränkung. Das Beispiel startet den IE9 aus beiden Umgebungen: „Start“ c:\progra~1\Internet Explorer\iexplore.exe“ www.google.de“ (64-Bit-Umgebung) und „Start“c:\progra~2\Internet Explorer\iexplore.exe“ www.google.de“ (32-Bit-Umgebung).

**Tipp 13: Farb-Auswahl und „Look & Feel“**

Die Farb-Auswahl „swan“ ist in einigen Dokumenten nicht aufgeführt und macht einen brauchbaren Eindruck. Listing 7 zeigt einen Vorschlag zu einem netten Layout.

```
webforms.cfg:
# colorScheme=teal, blaf, swan
colorScheme=swan
# lookandfeel=Generic, Oracle
lookandfeel=Oracle
```

Listing 7: Farbauswahl „swan“

**Fazit**

Mit der Version 11.2.0 hat Oracle die WebLogic-basierte Lösung von Forms/Reports gegenüber 11.1.1.x deutlich verbessert. Der Aufwand einer Migration kann nur durch eine sorgfältige Analyse der alten Software und eine möglichst präzise Konzeption der neuen Lösung bestimmt werden. Zu beachten sind hier besonders die bisherigen Schnittstellen und Office-Automatationen. Dabei sollte auch der Aufwand für die Konfiguration von Entwickler-, Anwender- und Server-Installation nicht vernachlässigt werden. Alle neuen

Technologien müssen exakt getestet, optimiert und mit dem Anwender abgestimmt werden, um einer so robusten und unverwüstlichen Client-Software wie Oracle Forms 6i ebenbürtig zu sein und beim Anwender nicht das Gefühl aufkommen zu lassen, die teure Migration wäre im Vergleich zur alten Lösung nur ein Rückschritt.

Frank Christian Hoffmann  
Cologne Data GmbH  
fch@cologne-data.de



## *Wir begrüßen unsere neuen Mitglieder*

**Firmenmitglieder**

Fabian Klose	Steltix Deutschland GmbH
Claudia Rosenberger	usb GmbH
Klaus Blaschek	b.telligent GmbH & Co. KG
Harald Sellmann	its-people Frankfurt GmbH

**Persönliche Mitglieder**

Kai Liegl	Volker Sikora
Tobias Arnhold	Gabriele Herold
Hermann von Kobylinski	Reinhard Hellenkamp
Zekeriya Kalabalik	Jörg Kröger
Corinne Krüger	Peter Schubert
Stefanie Baudach	Matthias Korn
Enrico Lausch	Patrick Schwanke
Claudia Romanova	Joachim Hofmann