



Arbeiten mit JDeveloper – Logische Konzepte

„Contrariwise,“ continued Tweedledee, „if it was so, it might be; and if it were so, it would be; but as it isn't, it ain't. That's logic.“

—Lewis Carroll (1832–1898), *Through the Looking Glass*

Die Java-Umgebung ist umfangreicher und komplexer als die traditionelle Entwicklungsumgebung. Begriffe wie Web-basierter Einsatz, Java Foundation Classes, Java-Server Pages, Einkapselung, Java Virtual Machine, Abstract Windowing Toolkit, Einkapselung, Common Object Request Broker Architecture, HTML, Polymorphismus, XML, Application Servers, Java-Datenbankkonnektivität, JARs, Swing, SQLJ, Applets, Servlets, JavaBeans, Multi-Tier und Enterprise zeigen die gegebenen Möglichkeiten und können gleichzeitig etwas entmutigend sein. Dennoch unterstützt Sie Oracle JDeveloper beim Aufbau, der Fehlersuche und dem Einsatz von Java-Geschäftskomponenten und Datenbankanwendungen für das Internet.

Dieses Kapitel bietet einen Überblick über die logischen Konzepte, die Sie bei der Entwicklung einer Anwendung mit JDeveloper parat haben sollten. Zur Verdeutlichung der Konzepte schließt das Kapitel mit einer praktischen Übung, die zeigt, wie man eine einfache und funktionierende Java-Anwendung schreibt, die mit der Datenbank interagiert.

1.1 Überblick

Wie in der Einführung bereits erwähnt, beschreibt dieses Buch nicht alle Möglichkeiten von JDeveloper. Schwerpunktmäßig soll es Entwickler von Client-Server-Applikationen unterstützen, die mit Produkten wie Oracle Forms, Visual Basic oder PowerBuilder arbeiten und den Umstieg auf Java und JDeveloper ins Auge fassen. Was

müssen Sie noch lernen? Gestaltet sich der Aufbau von Systemen in Oracle JDeveloper grundlegend anders als mit den bisherigen Werkzeugen? Nachdem die ersten Unklarheiten beseitigt sind, werden Sie feststellen, dass es doch nicht so viele Unterschiede gibt, wie zuerst angenommen. So spricht man beispielsweise bei der Entwicklung einer Lohnbuchhaltung in einer traditionellen Umgebung wie Oracle Form Developer nicht von einem „Applikationssystem“. Sie entwickeln eine Anzahl (vielleicht Hunderte) kleinerer Anwendungen, die sich gegenseitig koordinieren und rufen. Zusammen bilden sie das Applikationssystem. Und auch in JDeveloper gibt es kein „Applikationssystem“. Sie entwickeln weiterhin eine Anzahl kleinerer Anwendungen, die zusammenpassen.

Der erste wichtige Unterschied zwischen den traditionellen Umgebungen und JDeveloper ist, dass in manchen Entwicklungsumgebungen die Benutzerschnittstelle und die Interaktion mit der Datenbank untrennbar miteinander verbunden sind. In der JDeveloper-Umgebung sind Benutzerschnittstelle (UI) und die Interaktion mit der Datenbank getrennt. In JDeveloper legen Sie Dateien in logischen Containern, so genannten *Projekten*, an. Jedes Programm umfasst üblicherweise zwei Typen von JDeveloper-Projekten.

- **Komponenten und Logik der Benutzerschnittstelle** Sie werden üblicherweise mithilfe von Java und Java-Komponenten geschrieben.
- **Geschäftskomponenten für Java (BC4J)** Sie werden normalerweise mit Java und XML geschrieben und bieten die Komponenten zur Interaktion mit der Datenbank.

Bei dieser Architektur werden die meisten UI-Projekte nicht tatsächlich mit der Datenbank kommunizieren. Stattdessen wird die UI-Kommunikation mit der Datenbank von der mittleren Ebene, den BC4J-Objekten, ausgeführt. Das soll allerdings nicht heißen, dass ein UI-Projekt keinen Code für die Java Database Connectivity (JDBC) enthalten kann, doch die primären GUI-Objekte kommunizieren mit der Datenbank üblicherweise über BC4J-Objekte.

1.1.1 JDeveloper

JDeveloper ist eine Umgebung zur Anwendungsentwicklung, die die ersten Schritte in der Java-Welt erleichtert. Für den, der alles selbst macht, ist sie wie ein leeres Stück Papier; wer hingegen 4GL-Techniken wie Drag and Drop bevorzugt, für den agiert JDeveloper als Codegenerator. JDeveloper generiert auch den elementaren Code für die Datenbankschnittstelle, den Sie an Ihre speziellen Anforderungen anpassen.

Die ersten Schritte mit JDeveloper sind ungefähr so, als ob man Englisch und gleichzeitig den Umgang mit einer Textverarbeitung lernt. Bevor Sie sich auf dieses Abenteuer einlassen, sollten Sie bereits Erfahrung mit anderen Programmiersprachen und

Anwendungskonventionen gesammelt haben. Wichtig ist, die Punkte und Strukturen in den Praxisbeispielen nachzuvollziehen und auch selbst zu entwickeln. Die praktischen Erfahrungen aus diesen Beispielen helfen, sich recht schnell mit der Entwicklungsumgebung vertraut zu machen. Je nachdem, wie erfolgreich Sie sind, lernen Sie auch Möglichkeiten kennen, die im Bereich Troubleshooting und Debugging angeboten werden.

Beim Aufbau von Internet-Applikation setzen Sie wahrscheinlich nicht nur Java, sondern auch HTML-Code ein, der möglicherweise um JavaScript-Elemente erweitert ist. Diese Sprachen sind nicht Gegenstand dieses Titels; eine kurze Einführung zu diesen Themen findet sich in Kapitel 8.

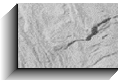
1.1.2 Der Multi-Tier-Ansatz bei der Anwendungsentwicklung

JDeveloper wurde dahingehend optimiert, dass es den Aufbau einer mehrschichtigen Architektur für Ihre Datenbankanwendungen unterstützt. Diese Architektur unterstützt die logische Trennung folgender Elemente:

- **Client-Ebene** Die Client-Ebene unterstützt die Anforderungen des Endbenutzers für den Datenzugriff. Komplexe GUIs lassen sich per Applikationscode über das Internet unterstützen. Dazu wird der Code auf dem Client-Desktop installiert und außerhalb des Browsers ausgeführt.
- **Anwendungsebene** Auf der Anwendungsebene wird die Geschäftslogik ausgeführt und der Datenzugriff koordiniert. Application Server enthalten die Geschäftslogik, die sich über zahlreiche Clients und Anwendungen hinweg wiederverwenden lässt, was die Möglichkeit eröffnet, über mehrschichtige Architekturen hinweg gemeinsam die physischen und logischen Ressourcen zu nutzen. Die Anwendungsschicht kann auch spezialisierte Reporting- und Analyse-Tools zur Behandlung komplexer Anforderungen von Seiten der Geschäftslogik enthalten.
- **Datenbankebene** In dieser Ebene werden die Daten gespeichert und abgefragt. Diese Schicht enthält auch Links auf andere externe Datenquellen und Anwendungen, die von Teilen der allgemeinen Architektur benötigt werden.

1.2 Praxisbeispiel: Aufbau einer Java-Applikation mithilfe von Assistenten

Nachdem Sie eine ungefähre Vorstellung von der Arbeit mit JDeveloper haben, ist es höchste Zeit, das Ganze einmal auszuprobieren. Das folgende Beispiel zeigt, wie man JDeveloper als Code-Editor einsetzt. In den folgenden Kapiteln führen Sie dieses Praxisbeispiel weiter und sollten daher wissen, wie die Benutzerschnittstelle des JDeveloper funktioniert. Die Benutzerschnittstelle von JDeveloper (die integrierte Entwicklungsumgebung oder IDE) wird in Kapitel 2 besprochen.



Hinweis:

*Nach dem ersten Öffnen von JDeveloper sehen Sie sich die Begrüßungsseite an (soll die Seite erneut geladen werden, wählen Sie **Help | Welcome Arbeitsbereich (Sample)**). Diese Seite bietet zahlreiche Links auf nützliche Dokumente und Online-Ressourcen, die den Start erleichtern. Wie in der Einführung bereits ausgeführt, bieten Benutzergruppen, Newsgroups und Firmenwebseiten viele gute Anhaltspunkte.*

In diesem ersten Praxisbeispiel entwickeln Sie mit den Assistenten in JDeveloper eine Java-Applikation. Diese Java-Anwendung zeigt über eine einfache Master-Detail-Beziehung Informationen zu Angestellten an. Die Daten für diese Anwendung stammen aus dem Schema Scott, das zur Start-Datenbank gehört und zusammen mit Oracle ausgeliefert wird. Sollten Sie keinen Zugriff auf diese Tabelle haben, setzen Sie sich bitte mit Ihrem DBA in Verbindung oder installieren Sie sie von der Installations-CD für Oracle.

Das Praxisbeispiel besteht aus folgenden Phasen:

1. Einrichtung des Arbeitsbereichs
2. Anlegen eines Projekts mit Geschäftskomponenten
 - Anlegen einer Projektdatei
 - Anlegen einer Verbindung
 - Vervollständigen des Business Components Project Wizard
3. Anlegen eines Projekts für Benutzerschnittstellen
4. Definieren eines Master-Detail-Datenformulars
 - Starten des Business Components Data Form Wizard
 - Auswahl der Attribute

- ❑ Hinzufügen eines Application Wrapper
- ❑ Kompilieren und Ausführen der Anwendung

Wenn Sie die Java-Applikation ausführen, sieht sie ähnlich wie in Abbildung 1-1 aus.

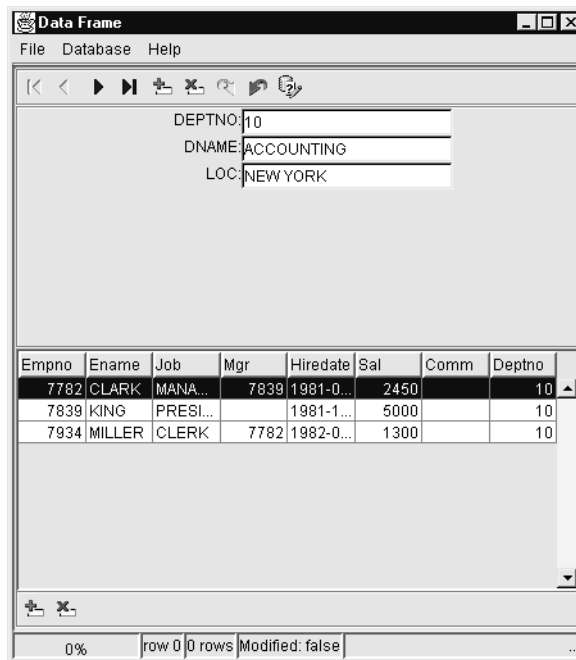
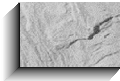


Abbildung 1-1: Die Java-Applikation.

1.2.1 Einrichten des Arbeitsbereichs

Vor der Entwicklung des eigenen Codes sind alle offenen Arbeitsbereiche zu schließen und ein neuer Arbeitsbereich anzulegen.

Arbeitet man mit JDeveloper, spielt sich alles in Arbeitsbereichen ab. Ein *Arbeitsbereich* ist ein logisches Konstrukt, das alle Elemente einer Anwendung aufnimmt. Ein Arbeitsbereich kann eine beliebige Anzahl von Projekten enthalten; erfahrungsgemäß sollten Sie jedoch für jede wichtige Bildschirmapplikation (was einem Formular in Oracle Developer oder einem Visual-Basic-Projekt entspricht) einen eigenen Arbeitsbereich vorhalten.

**Hinweis:**

*Sie können immer nur einen Arbeitsbereich geöffnet haben.
Diese produktspezifische Beschränkung wird in zukünftigen
Versionen hoffentlich beseitigt.*

Ein Arbeitsbereich ist physisch nichts anderes als eine Textdatei (mit der Erweiterung `.jws`), die Informationen über die Projekte und Anwendungen enthält. In diesem Praxisbeispiel legen Sie einen Arbeitsbereich namens `EmployeeWS` an, der die Dateien und Einstellungen protokolliert, die Sie bei der Entwicklung Ihrer ersten Java-Applikation einrichten beziehungsweise setzen. Die Arbeitsbereichsdatei wird zusammen mit den projektbezogenen Dateien in einem oder mehreren Ordnern gespeichert (Systemverzeichnisse). In der Entwicklungsphase bezeichnet man die physischen Unterverzeichnisse/Verzeichnisse (Pfade), in denen die Gruppen von zusammengehörigen Dateien gespeichert sind, als *Package*. Der Begriff *Package* verweist auch auf eine duplierte Struktur in einer Klassenbibliothek (`.jar` oder `.zip`). Das *JAR* (Java-Archiv) besteht aus einer komprimierten Datei mit dem kompilierten Code, den Sie einsetzen oder gemeinsam nutzen möchten, und einer zusätzlichen Datei mit einer Beschreibung der *JAR*-Inhalte. Ihre Arbeitsbereichsdatei kann Informationen über Dateien verwalten, die in verschiedenen *Packages* gespeichert sind. Sobald Sie einen Arbeitsbereich speichern, werden alle offenen Dateien und Fenster aktualisiert und gesichert.

1. Sollten Sie einen bestehenden Arbeitsbereich geöffnet haben, wählen Sie **File | Close Arbeitsbereich**. Der Navigation-Bereich zeigt daraufhin einen neuen Standard-Arbeitsbereich namens „Untitled1.jws“.
2. Wählen Sie **File | Save Arbeitsbereich**.
3. Um für Ihre Anwendung ein neues Verzeichnis anzulegen, klicken Sie im File-Dialog auf das Symbol **Create New Folder** und nennen das neue Verzeichnis „myempapp“.
4. Doppelklicken Sie `myempapp`.
5. Um der Arbeitsbereichsdatei einen Namen zu geben, klicken Sie in das Feld *File* und benennen `Untitled1` in „EmployeeWS“ um (siehe Abbildung 1-2). Klicken Sie auf **Save**.
6. Wählen Sie **File | Save All**.

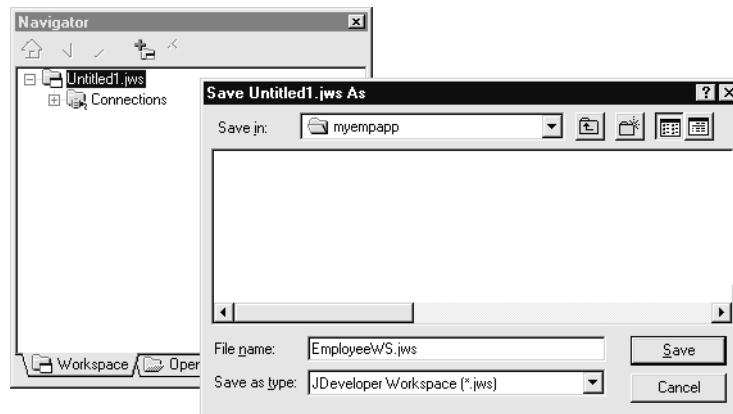


Abbildung 1-2: Anlegen eines Arbeitsbereichs.

Was ist gerade passiert? Sie legen einen Arbeitsbereich, einen Ordner beziehungsweise ein Package an, das die Arbeitsbereichsdatei und alle Dateien aufnimmt, die man für dieses Projekt benötigt.

Im nächsten Schritt werden die Metadaten für die Tabellen abgeholt, auf die Sie in der BC4J-Schicht zugreifen möchten. Dazu bietet JDeveloper den Business Components Project Wizard, der Sie bei dieser Aufgabe unterstützt. Die Anpassung der BC4J-Schicht besprechen wir in Kapitel 6. Im aktuellen Kontext sollten Sie diesen Code einfach als Gerüst betrachten, mit dessen Hilfe Sie Ihre Multi-Tier-Applikationen entwickeln, einsetzen und anpassen. In dieser Übung wird der Assistent für die Tabellen Employee und Department vordefinierte Views der Geschäftsdaten erstellen. Die BC4J-Schicht verwaltet für die Tabellen die Regeln der Parent-Child-Beziehungen und ist für die Aktualisierung der Datenbank verantwortlich, sobald Sie Zeilen einfügen, löschen oder ändern.

Der Quellcode (Klasse) ist in Dateien mit der Erweiterung `.java` zu hinterlegen. Jede Quelldatei trägt den gleichen Namen wie die Klasse, die sie definiert, und in einer Projektdatei wird jede der Quelldateien referenziert. Bei der Kompilierung speichert JDeveloper den neuen *Bytecode* (der sich in der Java Virtual Machine oder JVM ausführen lässt) in einer Datei, die den gleichen Namen wie die Quelldatei, aber die Erweiterung `.class` trägt. Kompilieren Sie ein Java-Programm, generiert man keine `.exe`-Datei und das Programm lässt sich auch nicht direkt rufen. Zur Ausführung Ihres Java-Programms starten Sie stattdessen die JVM und übergeben die Klassendatei als Parameter. Diese Anforderung ist kein unlösbares Problem, da die meisten modernen Betriebssysteme mit einer eigenen JVM ausgeliefert werden.

1.2.2 Anlegen eines Projekts mit Geschäftskomponenten

Innerhalb des Arbeitsbereichs sind Dateien in *Projekte* aufgeteilt. Die Signifikanz eines Projekts erschließt sich nicht so sehr auf Codeebene, sondern erfüllt eher die Funktion eines Gruppen-Containers für Dateien.

Physisch handelt es sich um eine Datei mit der Erweiterung `.jpr`. Bei BC4J-Projekten kommen die Erweiterungen `.jpr` oder `.JSP` in Frage. In dieser Phase der praktischen Übung legen Sie ein BC4J-Projekt an, das eine Liste mit Quelldateien (mit der Erweiterung `.java`) und Metadatendateien (mit der Erweiterung `.xml`) für die Schicht der Geschäftskomponenten enthält. Diese Trennung der Projekte unterstützt den Aufbau einer sauberen und gut strukturierten Applikation, wo ein Großteil des Datenbankzugriffs an einer Stelle eingekapselt wird.

Hinweis für Entwickler mit Oracle Developer

Das Handling des Datenbankzugriffs bei BC4J lässt sich mit dem vergleichen, wie die Blöcke in Oracle Forms auf die Datenbank zugreifen. Der Event-Code in der primären Anwendung entspricht den Triggern und Programmeinheiten, die in Forms auf die Datenbank zugreifen.

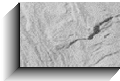
Bevor Sie in JDeveloper arbeiten, ist stets ein Arbeitsbereich und eine Projektdatei einzurichten. Diese Dateien dienen in der Entwicklungsphase der logischen Organisation des Codes und als Inhaltsangabe für die auszuführenden Arbeiten.

Sie können beliebig viele Projekte einrichten, die auf die gleichen Klassendateien zeigen. Zudem lassen sich die Projektdateien an jedem Standort hinterlegen, unabhängig von den Klassendateien, auf die sie zeigen. Dennoch sollten Sie bei der Hinterlegung des Java-Arbeitsbereichs, der Projekte und Klassendateien auch die Skalierbarkeit berücksichtigen. Sollten Sie alle Dateien an einem Standort hinterlegen (wie `JDEV_HOME\myprojects`), wird Ihr Root-Verzeichnis recht bald von Dateien überschwemmt sein. Deshalb sollten Sie Ihre Dateien in einer Verzeichnishierarchie abspeichern, wo der Arbeitsbereich, die Projektdateien und die dazugehörigen Klassendateien im gleichen Ordner liegen.

Anlegen einer Projektdatei

Beim Anlegen einer Projektdatei können Sie zusätzlich Assistenten ausführen. Damit legen Sie andere Applikationsdateien an, die in Ihrem Projekt referenziert werden.

1. Um die Begrüßungsseite des Projekt Wizard aufzurufen, wählen Sie **File | New Project**. Klicken Sie zur Anzeige der Project Type-Seite auf **Next**.

**Hinweis:**

JDeveloper-Assistenten, die aus mehr als einer Seite bestehen, starten mit einer Begrüßungsseite, in der Sie eine kurze Beschreibung zu den Aufgaben des Assistenten finden. Falls Sie die Seite beim nächsten Aufruf nicht sehen möchten, entfernen Sie die Markierung für die Option „Display this page next time“. Bei allen Übungen in diesem Buch nehmen wir an, dass diese Markierung entfernt wurde.

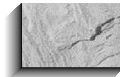
2. Zur Anzeige des Dialogs Select Page klicken Sie auf Browse. Mit einem Doppelklick auf myempapp öffnen Sie den Ordner/Package, in dem Sie Ihre Dateien speichern möchten.
3. In der Seite Project Type benennen Sie MyProjext1 in „MyDeptEmp-BusCompPRJ“ um und klicken auf Open. JDeveloper hängt an den Namen der Projektdatei die Erweiterung .jpr an und bringt Sie zur Seite Project Type zurück (Kapitel 3 bespricht alle Fragen im Zusammenhang mit den Benennungsregeln für Dateien, die Sie in JDeveloper anlegen).
4. Wählen Sie in der gleichen Seite unter der Überschrift *What type of project would you like to create?* den Eintrag „A Project Containing Business Components“ und klicken Sie auf Next.
5. In der Seite für die Projektoptionen klicken Sie neben *What is the name of the project's default package* auf Browse und starten damit den Package Browser. Wählen Sie „myempapp“ als Standardordner/Package für Ihr Projekt und klicken Sie auf OK. Alle anderen Standardwerte lassen Sie unverändert.
6. Zum Verlassen der Seite mit den Projektoptionen klicken Sie auf Next.
7. In der Seite für die Projektinformationen können Sie eine Beschreibung zum Projekt hinterlegen. Lassen die Einstellung für das Kontrollkästchen *Generate project HTML file* wie vorgegeben. Nach dem Eintragen der Beschreibung klicken Sie auf Next.
8. Jetzt sehen Sie eine Zusammenfassung der ausgewählten Optionen. Sehen Sie sich den Inhalt dieser Seite kurz an.
9. Mit einem Klick auf Finish legen Sie das Projekt an und rufen den Business Components Project Wizard. Sollte die Begrüßungsseite erscheinen, klicken Sie auf Next.

Aufbau einer Verbindung

Bei der Entwicklung einer Anwendung haben Sie üblicherweise Zugriff auf eine begrenzte Anzahl von Datenbankinstanzen. Der Verbindungsaufbau ist eine sich ständig wiederholende Aufgabe, die immer wieder auszuführen ist. Zur Erleichterung dieser

Operation hat JDeveloper eine Definition namens Connection implementiert, die von allen Entwicklungssitzungen genutzt wird. In der Definition werden die Benutzernamen, Passwörter, URLs und dazugehörigen Parameter für Ihre Datenbankverbindungen gespeichert. Um die bereits implementierten Connections zu sehen, öffnen Sie den Connection-Knoten (im Navigator-Bereich). Nach dem Doppelklick auf eine Verbindung erhalten Sie unmittelbaren Zugriff auf den Database Browser, der die Datenbankobjekte anzeigt.

1. In der Connection-Seite klicken Sie zum Öffnen des Connection-Dialogs auf New. Hier definieren Sie die einzusetzenden Datenbankverbindungen.



Hinweis:

Lassen Sie die Informationen für den Connection-Dialog von Ihrem DBA prüfen.

2. Füllen Sie die Felder wie folgt aus. Dabei wird angenommen, dass sich die Datenbank auf Ihrem PC befindet.

Connection Name ist „ScottConnectionAuto“

User ist „SCOTT“

Password ist „tiger“

Connection Type ist „JDBC“

Include password in deployment archive (markieren Sie dieses Kontrollkästchen)

JDBC Driver ist „Oracle JDBC Thin“

Connection method ist „Named Host“

Host ID ist „localhost“

SID ist „ORCL“

Port ist „1521“

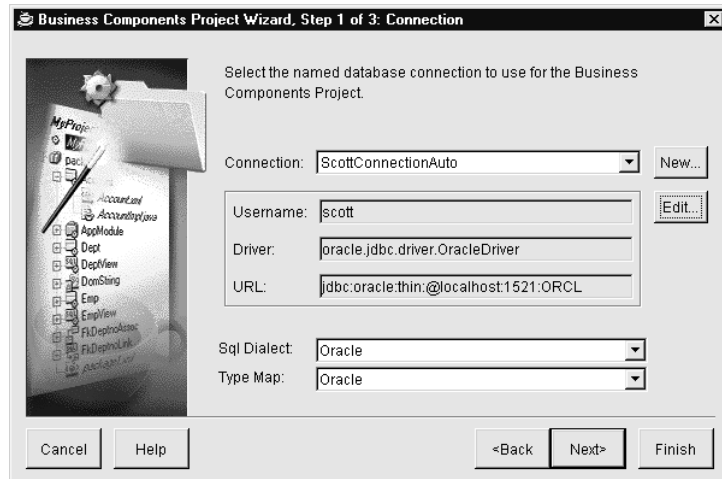
Row Prefetch ist „10“

Batch Value ist „1“

Report TABLE_REMARKS (lassen Sie dieses Kontrollkästchen unmarkiert)

3. Klicken Sie auf Test Connection. Wurden die richtigen Informationen eingegeben, sehen Sie die Nachricht „Success“, die auf der rechten Seite der Schaltfläche Test Connection eingeblendet wird.

Mit einem Klick auf OK beenden Sie den Connection-Dialog. Die Einstellungen erscheinen in der Connection-Seite. Klicken Sie auf Next.



Ist der Verbindungstest erfolgreich, öffnet sich die Seite Package Name. Scheitert die Verbindung, zeigt JDeveloper eine Fehlermeldung an und bleibt in der Connection-Seite. Im Falle einer Fehlermeldung aktualisieren und bestätigen Sie die Verbindungsinformation. Mit einem Klick auf Test Connection stellen Sie sicher, dass die Verbindung funktioniert. Erst wenn die Verbindung steht, macht der Assistent weiter.

Den Business Components Project Wizard beenden

Im letzten Abschnitt legten Sie eine Verbindung an. Diese Operation wird normalerweise nur einmal ausgeführt. Benötigen Sie diese Verbindung in späteren Projekten, wählt man diese aus der Auswahlliste in der Connection-Seite aus. Der folgende Schritt beendet die Definition der Geschäftskomponenten und generiert den Code.

1. Akzeptieren Sie in der Seite Package Name den Standardnamen myempapp. Klicken Sie auf Next.
2. In der Seite Business Components sollte das Schema-Feld „SCOTT“ zeigen und das Kontrollkästchen Tables sollte markiert sein. Wählen Sie im *Available*-Feld DEPT und EMP (mit Strg-Klick) aus und verschieben Sie die beiden Einträge mit der Schaltfläche > in das *Selected*-Feld.
3. Markieren Sie die Kontrollkästchen *View Objects* und *View Links and Application Module*. Klicken Sie auf Next.
4. Zum Anlegen der Geschäftskomponenten klicken Sie auf Finish. Die Komponenten werden in Ihr Projekt eingefügt.
5. Speichern Sie Ihre Arbeit mit **File | Save All**.

Was ist gerade passiert?

Mithilfe des Assistenten legten Sie ein BC4J-Projekt an, das sich mit einem Datenbankschema verbindet und die Metadaten für die Tabellen EMP und DEPT definiert. Damit erhalten Sie die gewünschten Datenelemente, die Sie in der nächsten Phase zum Anlegen der Benutzerschnittstelle benötigen.



Hinweis:

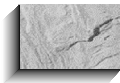
Sehen Sie sich den Code an, der von JDeveloper bisher generiert wurde. Jetzt haben Sie über 1000 Zeilen Quellcode, den Sie genau an Ihre Anforderungen anpassen können. Wenn Sie im Navigation-Bereich die Dateinamen doppelklicken, öffnen sich die Dateien im Source Editor. Sehen Sie sich auch die XML-Dateien mit den Definitionen für die BC4J-Objekte an. Ganz sicher finden Sie einige Begriffe, Tags, Specifier, Klassen und Sprachkonstrukte, die neu für Sie sind.

1.2.3 Anlegen eines Projekts für die Benutzerschnittstelle

Das folgende Projekt fügt dem Arbeitsbereich den GUI-Code hinzu, mit dem die Daten aus DEPT und EMP angezeigt werden. Dieses Projekt isoliert den GUI-Code logisch von dem der Geschäftskomponenten. Jedes Projekt repräsentiert einen anderen funktionalen Bereich, der jeweils eigenständig entwickelt wird. Projekte besitzen ein oder mehrere Einsatzprofile, die beschreiben, wie das Projekt einzusetzen ist. Kapitel 8 bespricht die Einsatzprofile und mögliche Alternativen.

Obwohl Sie die Dateien für die Applikation in einem zweiten Projekt hinterlegen, werden Sie feststellen, dass alle Quelldateien dieser Übung physisch am gleichen Ort liegen. Da wir in diesem Beispiel ein einziges Package (Unterverzeichnis) einsetzen, ist für jede Datei, die in diesem Unterverzeichnis liegen soll, ein eindeutiger Name anzugeben.

1. Zum Start des Project Wizard klicken Sie auf **File | New Project**. Um die Begrüßungsseite zu überspringen, klicken Sie auf **Next**.
2. In der Seite **Project Type** erhalten Sie nach einem Klick auf **Browse** den Dialog **Select Project** angezeigt. Um zum Ordner des Projekts zu gelangen, doppelklicken Sie „myempapp“.
3. Nennen Sie Ihr Projekt "EmpAppPRJ" und klicken Sie „open“.
4. In der Seite wählen Sie unter *What type of project would you like to create?* die Option *An empty project field* und klicken auf **Next**.

**Hinweis:**

Wenn Sie an dieser Stelle „A project containing a new“ wählen und zusätzlich „Application“ angeben, wird der Business Components Data Form Wizard gestartet. Da diese Vorgehensweise zahlreiche Schritte umgeht, zeigt dieser Ansatz nicht, welche Dateien eigentlich angelegt werden.

5. Zum Start des Package Browser klicken Sie neben *What is the name of the project's default package* auf Browse. Wählen Sie „myempapp“ und klicken Sie auf OK.
6. Verändern Sie bei den Dateinamen/Pfaden und den dazugehörigen Standardwerten nichts. Klicken Sie auf Next.
7. In der Seite mit den Projektinformationen können Sie eine Beschreibung hinterlegen. Markieren Sie das Kontrollkästchen für *Generate project HTML file* nicht. Klicken Sie auf Next.
8. Jetzt sehen Sie eine Übersicht mit den selektierten Optionen. Um das Projekt anzulegen, klicken Sie auf Finish. Funktioniert alles wie geplant, sehen Sie im Navigation-Bereich eine Projektdatei (EmpAppPRJ.jpr) mit Verbindungen.



9. Wählen Sie File | Save All.

Was ist gerade passiert? Sie legten ein neues Projekt mit dem GUI-Code und den Dateien für die Applikation an, die Sie gerade entwickeln.

**Hinweis:**

Aus Gründen der Vereinfachung speichern wir bei dieser Übung das BC4J-Projekt im gleichen Ordner wie die Anwendung. In der Praxis ist es allerdings besser, dem Projekt einen eigenen Ordner zuzuweisen. Das erleichtert die Wiederverwendung in anderen Projekten.

1.2.4 Definition eines Master-Detail-Formulars

Der Business Components Data Form Wizard führt jetzt schrittweise zum Anlegen der Benutzerschnittstelle. Diese Schnittstelle basiert auf Informationen der Tabellen DEPT und EMP, die in einer Master-Detail-Beziehung verknüpft sind. Für den Zugriff auf die Daten nutzt sie die Datenbankschnittstelle, die im Projekt mit den Geschäftskomponenten entwickelt wurde. Nach Auswahl der benötigten Optionen generiert der Assis-

tent den Java-Quellcode, den man zur Implementierung der Schnittstelle in einem Java-Gerüst benötigt. Später entwickeln Sie eine Applikationsklasse, die dieses Gerüst zur Anzeige des Geschäftsformulars ruft.

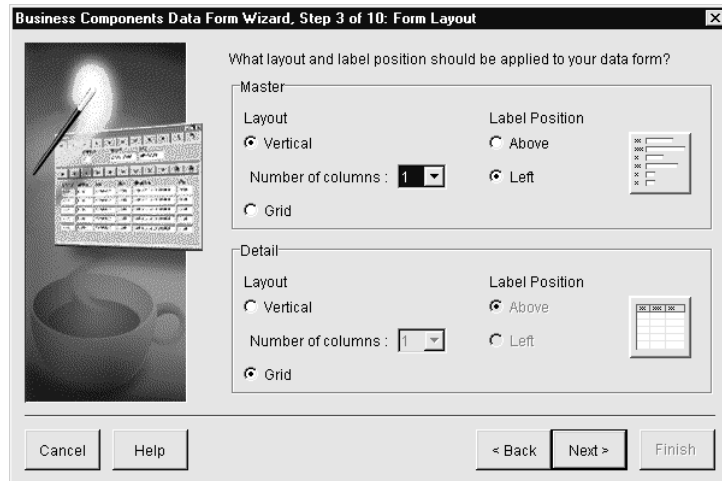
Start des Business Components Data Form Wizard

Die folgenden Schritte starten den Assistenten, der im UI-Projekt eine Frame-Klassen-datei anlegt.

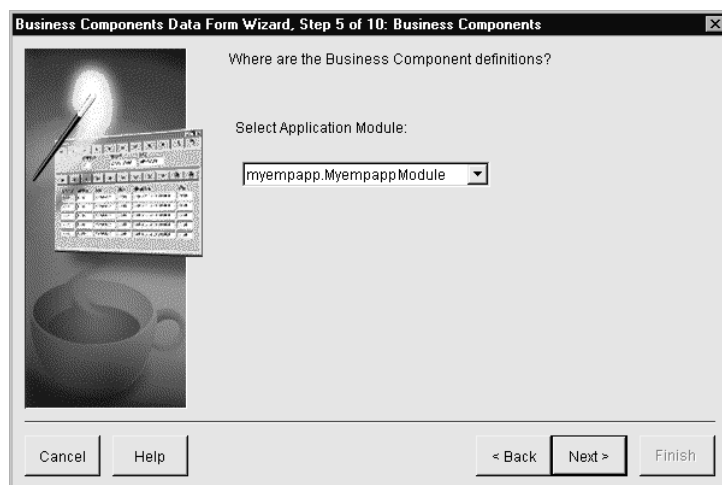
1. Wählen Sie im Bereich Navigation EmpAppPRJ.jpr. Zur Anzeige der Object Gallery wählen Sie **File | New**.
2. Wählen Sie das Symbol Business Components Data Form. Sie sehen die Begrüßungsseite des Business Components Data Form Wizard.
3. Mit einem Klick auf Next gelangen Sie zu der Seite, in der Sie ein Formular definieren können.
4. Wählen Sie den gewünschten Formulartyp. In unserem Fall ist es „Master-Detail“.
5. Als Implementierungspräferenz wählen wir „Frame“. Nach einem Klick auf Next sehen Sie die Frame-Seite.
6. Als Namen für den Frame geben wir „MyEmpFrame“ ein. Die Standardvorgaben für die Felder *Package* und *Title* werden übernommen. Klicken Sie auf Next. Sie erreichen die Seite Form Layout.

Der Business Components Data Form Wizard bietet für das Layout der Formulare verschiedene UI-Formatvorlagen an. In dieser Übung zeigen wir die Inhalte der Department-Spalten im Formularkopf und einzelnen Feldern an. Die Employee-Spalten erscheinen im Hauptteil des Formulars und werden in ein Raster gepackt. Die Formatvorlage „Frame with Navigation Bar“ erfüllt unsere Vorgaben und wird das Erscheinungsbild des Formulars verwalten.

7. Für die Master-Tabelle wählen Sie als *Layout* „Vertical“. Für die Felder selektieren Sie bei *Label Position* jeweils „Left“.
8. Für die Detail-Tabelle wählen Sie beim *Layout*-Feld „Grid“. Achten Sie darauf, dass *Label Position* auf „Above“ gesetzt ist. Die Seite sollte wie folgt aussehen:



9. Zur Anzeige der Template-Seite klicken Sie auf Next.
10. Wählen Sie die Formatvorlage „Frame with Navigation Bar“ und klicken Sie auf Next. Sie erreichen die Seite Business Components.

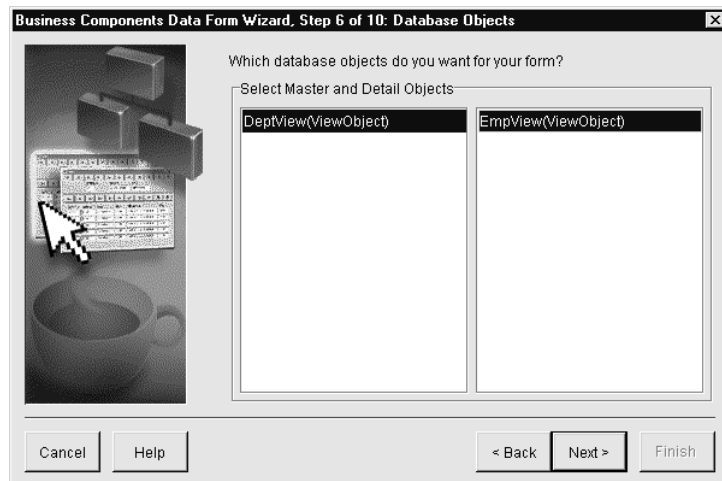


11. Akzeptieren Sie das Standard-Applikationsmodul und klicken Sie auf Next. Sie sehen die Seite Database Objects.

Jetzt legen Sie zwischen den Objekten DeptView und EmpView, die im Projekt mit den Geschäftskomponenten generiert wurden, eine Master-Detail-Beziehung an. Diese View-Objekte bieten den Zugriff auf Daten des Schemas Scott.

12. Zur Auswahl des Master-Objekts klicken Sie auf DeptView(ViewObject).

13. Zur Auswahl des Detail-Objekts klicken Sie auf EmpView(ViewObject). Die Seite sollte wie folgt aussehen:

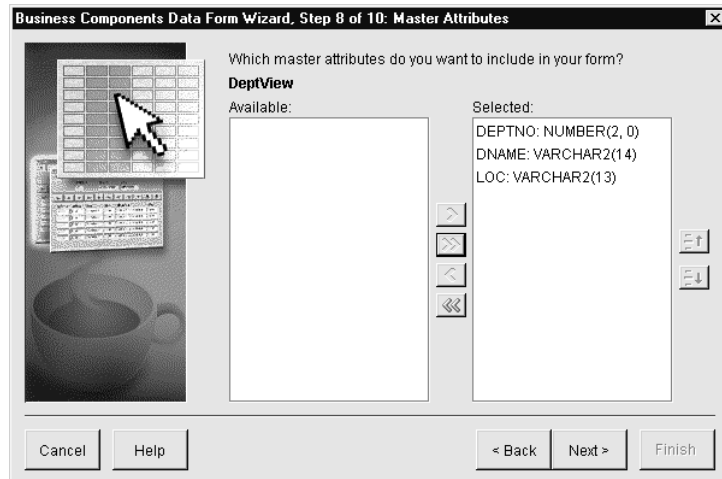


14. Mit einem Klick auf Next gelangen Sie zur Seite Join Condition. Die standardmäßige Join-Bedingung dient der Synchronisierung der Employee-Datensätze, wenn der Benutzer die Department-Datensätze durchblättert.
15. Stellen Sie sicher, dass beide Views markiert sind. Mit einem Klick auf Next akzeptieren Sie die Verknüpfung.

Attribute wählen

Der Assistent führt alle Attribute der Master- und Detail-View-Objekte auf. Diese Attribute repräsentieren die Datenbankspalten in DEPT und EMP. Die Informationen hinter den Attributnamen zeigen den Datentyp und die Länge der Spalte in der Tabelle. Mit folgenden Schritten wählen Sie die Spalten für das Formular.

1. Klicken Sie in der Seite Master Attributes auf die Schaltfläche >>. Mit dieser Operation verschieben Sie alle verfügbaren Attribute in die *Selected*-List.

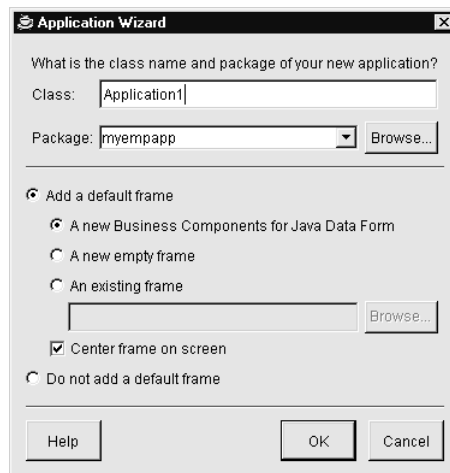


2. Mit einem Klick auf Next gelangen Sie zur Seite Detail Attributes.
3. Um alle Attribute von der *Available*- in die *Selected*-Liste zu verschieben, klicken Sie auf >>.
4. Klicken Sie auf Next. Sie gelangen zur Seite Deployment Connection. Die bereits weiter oben eingerichtete Verbindung (ScottConnectionAuto) sollte bereits ausgewählt sein.
5. Klicken Sie auf Next, um zur Seite Finish zu gelangen. Es wird eine Zusammenfassung der in den Wizard eingegebenen Definitionen angezeigt.
6. Um den Rahmen und den UI-Code für das Geschäftsformular zu generieren, klicken Sie auf Finish.
7. Nach dem Entwurf der Benutzerschnittstelle speichern Sie Ihre Arbeit über **File | Save All**.

Einfügen eines Application Wrapper

Das letzte Codeelement, das Sie für Ihr UI-Projekt generieren, ist eine Klasse namens *Application*. Diese Klasse repräsentiert das ausführbare Programm zur Anzeige Ihres Frames. Mithilfe des Applikation Wizard generieren Sie eine Klasse, die die Bildschirmgröße prüft und dann den Frame in einem Fenster mit passender Größe anzeigt.

1. Klicken Sie im Bereich Navigation auf EmpAppPRJ.jpr.
2. Zur Anzeige der Object Gallery wählen Sie **File | New**.
3. Wählen Sie im Register Objects das Symbol Application. Klicken Sie auf OK. Der Application Wizard wird wie folgt angezeigt:



4. Ändern Sie *Class* in „MyEmpApp“ und akzeptieren Sie für das *Package*-Feld die Standardeinstellung (myempapp).
5. Unter *Add a default frame* klicken Sie auf „An existing frame“. Zum Öffnen des Class Browsers klicken Sie rechts in das leere Feld.
6. Im Class Browser klicken Sie links neben myempapp auf das Symbol + . Das öffnet den Knoten. Mit einem Klick auf MyEmpFrame wählen Sie den zuvor angelegten Frame für Ihre UI-Komponenten.
7. Mit einem Klick auf OK beenden Sie den Class Browser und kehren zum Application Wizard zurück. Lassen Sie das Kontrollkästchen *Center frame on screen* markiert. Ein Klick auf OK beendet den Assistenten und stößt die Generierung des Codes an.
8. Speichern Sie Ihre Arbeit über **File | Save All**.

Anwendung kompilieren und ausführen

Jetzt kompilieren Sie sämtliche Projekte, die Sie zur Ausführung Ihrer Java-Anwendung vorbereitet haben. Bei der Kompilierung der Projekte generiert JDeveloper unter ihrem Ausgabeverzeichnis (JDEV_HOME\myclasses) eine duplierte Verzeichnisstruktur und hinterlegt dort die kompilierten Bytecode-Dateien mit einer .class-Erweiterung.

Mit den folgenden Schritte kompilieren Sie die Projekte für Ihre Java-Anwendung:

1. Im Bereich Navigation klicken Sie auf MyDeptEmpBusCompPRJ.jpr und wählen danach **Project | Rebuild Project „MyDeptEmpBusCompPRJ.jpr“**.
2. Klicken Sie auf EmpAppPRJ.jpr und wählen Sie danach **Project | Rebuild Project „EmpAppPRJ.jpr“**.
3. Sichern Sie die Auswahl über **File | Save All**.

4. Klicken Sie im Navigation-Bereich unter EmpAppPRJ.jpr auf die Datei MyEmpApp.java.
5. Wählen Sie **Run | Run „MyEmpApp“**. Akzeptieren Sie den eingeblendeten Copyright-Vermerk.
6. Klicken Sie im Login-Dialog auf OK. Falls notwendig, geben Sie ein Passwort ein.
JDeveloper kompiliert Ihren Quellcode. Werden keine Fehler generiert, führen Sie die Anwendung aus. Nehmen Sie sich Zeit und sehen Sie sich die Charakteristiken der Applikation etwas genauer an. Die Anordnung der Spalten können Sie mit Drag-and-Drop ändern; die Zeilen sortieren Sie mit einem Mausklick auf die Überschriften der verschiedenen Spalten; über die Auswahl des Spaltennamens im Kontextmenü (Rechtsklick) lassen sich die Spalten anzeigen und verbergen; auch die Größe des Fensters lässt sich ändern.

Was ist gerade passiert? Sie haben gerade Ihre erste Java-Anwendung mit JDeveloper fertig gestellt und dabei fast 40 Seiten Quellcode generiert. Das ist eine kleine Meisterleistung, aber selbst für die kleinste Geschäftsanwendung bleibt noch jede Menge Arbeit zu tun. Sie werden sehr schnell herausfinden, dass der Assistent nicht alle Wünsche erfüllt. Dennoch ist das kein großes Problem, da sich in dieser flexiblen Umgebung quasi jede Zeile generierten Codes bearbeiten lässt.