

Palm-Datenbanken Synchronisation (Conduits)

29.11.2001

Michael Dyrna
dyrna@in.tum.de

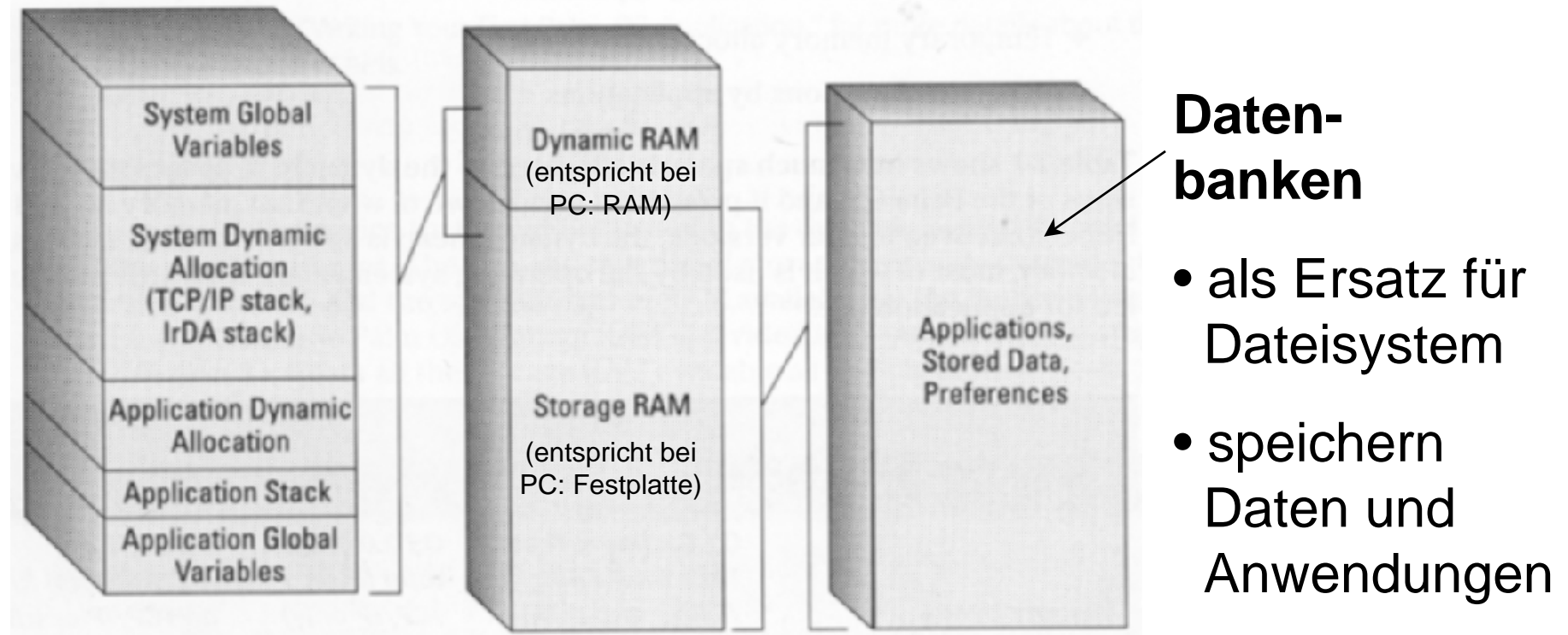
Übersicht

Teil I

Palm-Datenbanken

- Einführung: Wozu Datenbanken?
- Aufbau im Speicher
- Ressource- und Record-Datenbanken
- Datenbank-Header & Strukturen
- Preferences-Databases
- Feature Memory
- Data Manager API
- Fragmentierung

Wozu Datenbanken?

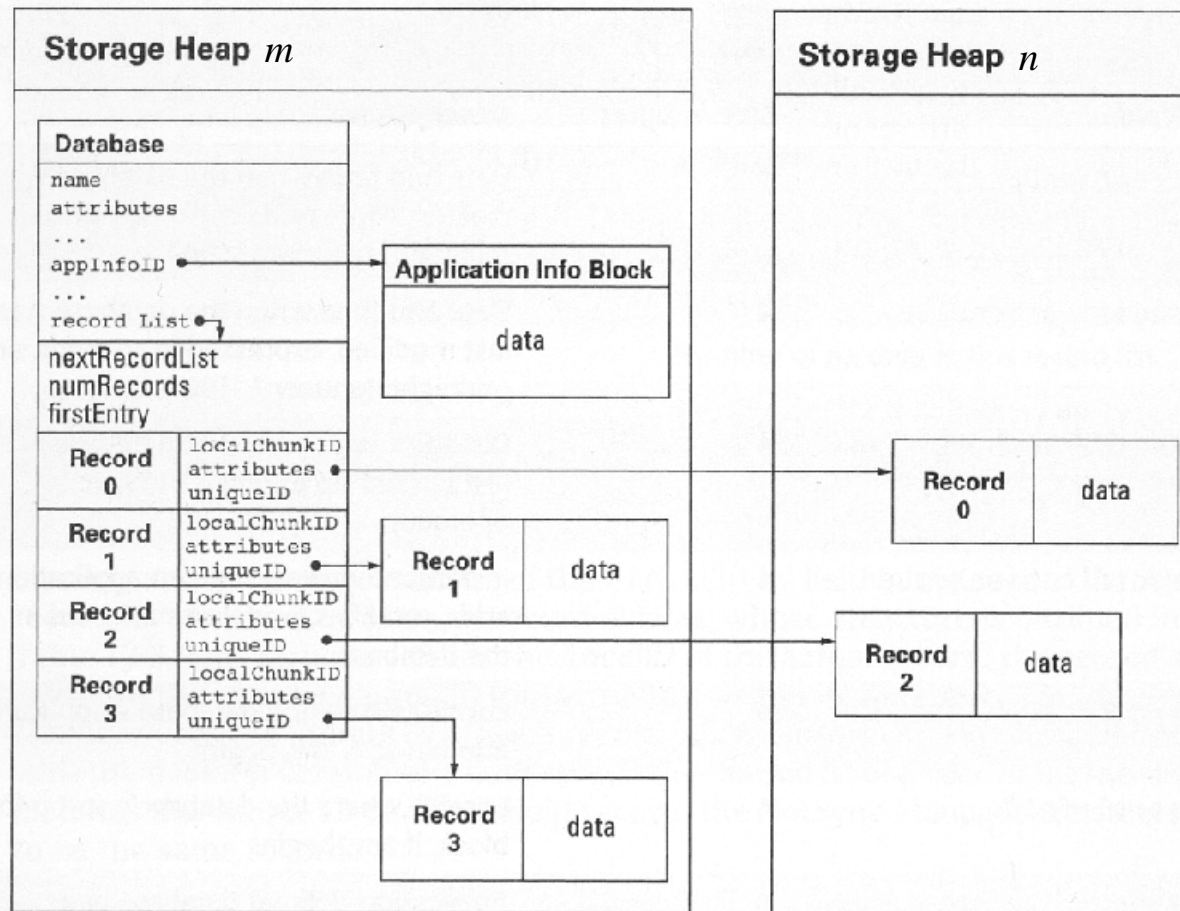


Quelle: Foster: Palm OS Programming Bible

Zugriff nur über Data Manager API!

Palm-Datenbanken

Palm-Datenbank = Liste von Speicherblöcken (*chunks*) im Storage RAM plus Header



- Chunks können in unterschiedl. Heaps liegen
- LocalID = Speicheradresse (kann sich verändern)

Quelle: Foster: Palm OS Programming Bible

Ressource-/Record-Datenbanken

zwei Arten von Datenbanken:

Ressource-Datenbank

- kein festes Format
- Datensätze variabler Länge
- keine Reihenfolge

- Code, Daten und Oberflächenelemente von Programmen
- Ersatz für Dateisystem

Record-Datenbank

- festes Format
- Datensätze fester Länge
- Reihenfolge

- “typische” Art Datenbank
- Benutzerdaten (z.B. Termine, Mails, Adressen, Aufgaben, Bookmarks, Webseiten)

Datenbank-Header

Feld	Länge	Beschreibung
name	32	Name der Datenbank
attributes	2	Eigenschaften
version	2	Versionsnummer des Formats
creationDate	4	Erstellzeit (in Sekunden seit 1.1.1904)
modificationDate	4	Zeit der letzten Änderung
lastBackupDate	4	Zeit des letzten Backups
modificationNumber	4	Zähler, der jedes Mal inkrementiert wird, wenn ein <i>record</i> erstellt, verändert oder gelöscht wird
appInfoID	4	<i>LocalID</i> , wo der <i>application info block</i> der Anwendung beginnt
sortInfoID	4	<i>LocalID</i> , wo der <i>sort info block</i> beginnt
type	4	Typ (von Anwendung definiert)
creator	4	<i>CreatorID</i> der Anwendung
uniqueIDSeed	4	vom System benutzt, um eine eindeutige ID-Nummer für jeden <i>record</i> zu erstellen
recordList	4	<i>LocalID</i> der ersten <i>record list</i>

application info block: enthält Kategorie-Namen und andere Informationen über die Datenbank

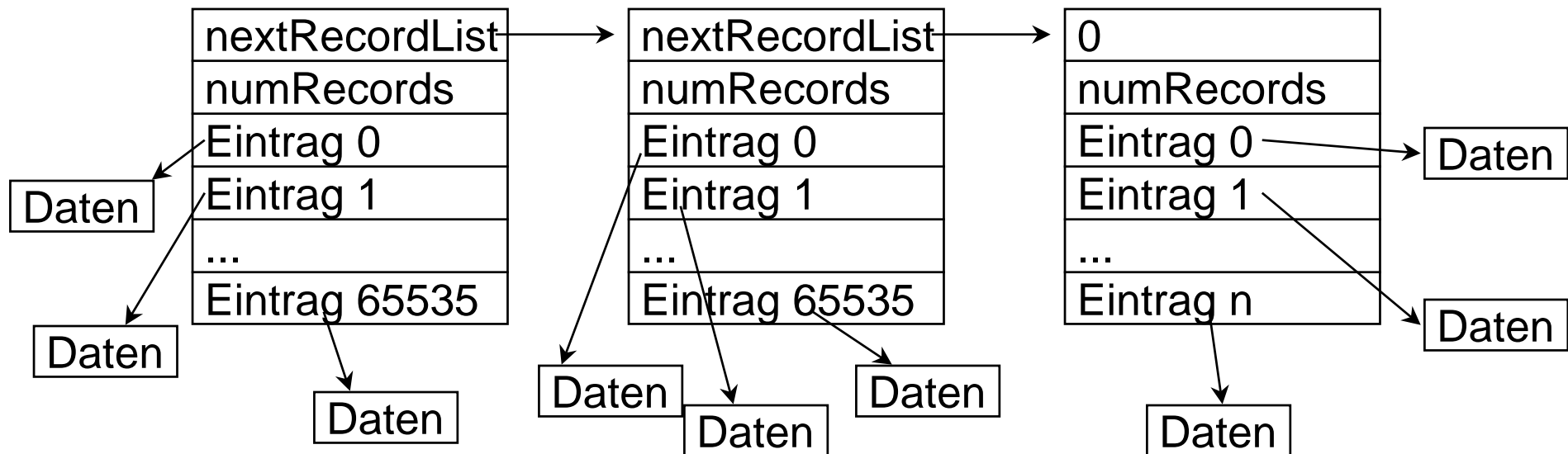
record = Datenbankeintrag

LocalID = Speicher-Adresse

sort info block: Liste von *record*-Nummern in einer bestimmten Reihenfolge

Record List Struktur

Feld	Länge	Beschreibung
nextRecordListID	4	LocalID der nächsten <i>record list</i> (0, falls keine weitere <i>record list</i>)
numRecords	2	Anzahl <i>records</i> in dieser Liste
Einträge...		<i>record</i> -Beschreibungen (unterschiedliches Format für Record- und Ressource-Datenbanken)



Record List Element Struktur

für Record-Datenbank:

Feld	Länge	Beschreibung
localChunkID	4	<i>LocalID</i> des record
flags	1/2	Flags: <i>secret, busy, modified, delete</i>
category	1/2	Kategorie
uniqueID	3	Eindeutige <i>ID</i> für diesen record

für Resource-Datenbank:

Feld	Länge	Beschreibung
type	4	Ressourcen-Typ
id	2	ID der Ressource (eindeutig pro Typ)
localChunkID	4	<i>LocalID</i> der Ressource

Andere Möglichkeiten der Datenspeicherung

“Preferences Database”:

- zur Speicherung von einfachen Variablen zwischen zwei Programmaufrufen ist eigene Datenbank übertrieben
=> PalmOS verwaltet eine Datenbank für die Eigenschaften von allen Programmen

“Feature Memory”:

- ab PalmOS 3.1
- in Storage RAM (d.h. persistent)
- erlaubt “unstrukturierten” Speicherzugriff (byte- und blockweises Lesen und Schreiben von Daten)
- entspricht Dateizugriff beim PC

Data Manager API (Übersicht)

Operationen auf Datenbanken

DmCreateDatabase
DmOpenDatabase
DmOpenDatabaseByTypeCreator
DmCloseDatabase
Dm(Set)DatabaseInfo
DmFindDatabase
DmDeleteDatabase

Operationen auf Records

DmNewRecord
DmMoveRecord
DmRemoveRecord
Dm(Set)RecordInfo
DmQueryRecord
Dm[Insert|Quick]Sort
DmFindSortPosition

Operationen auf Feature Memory

FtrPtr[New|Free]
DmWrite
FtrGet

Operationen auf Preferences

Pref[Get|Set]AppPreferences

Speicherfragmentierung

- PalmOS 1.0:
- mehrere Heaps à 64 KB
 - Belegungen werden so verteilt, dass alle Heaps ungefähr gleich groß sind
 - => große Speicheranforderung kann nicht erfüllt werden, obwohl viele kleine freie Bereiche
- PalmOS 2.0:
- Speicher wird immer in dem Heap belegt, in dem am meisten Platz frei ist
 - => Problem wie bei 1.0
- PalmOS 2.0.4:
- kann eine Speicheranforderung nicht direkt erfüllt werden, werden Speicherblöcke verschoben
- PalmOS 3.0:
- ein großer Heap für alle Datenbanken

Zusammenfassung (Palm-Datenbanken)

- Ein Zugriff auf das Storage RAM ist nur über die Data Manager API möglich
- Eine Palm-Datenbank besteht aus Header-Informationen und einer Liste von Records oder Ressourcen.
- Datenbanken werden anhand der LocalID adressiert.
- Jede Datenbank kann mit ihrem eindeutigen Namen, der CreatorID und ihrem Typ identifiziert werden.
- Der Application Info Block enthält Informationen über die ganze Datenbank.
- Programmeigenschaften werden von Palm OS in einer eigenen Datenbank verwaltet.

Übersicht

Teil II

Synchronisation & Conduits

- Einführung: Wozu Synchronisation?
- mögliche (Hardware-)Schnittstellen
- Conduits
- Synchronisations-Vorgang
und Mechanismen
- Entwicklung von Conduits

Wozu Synchronisation?

komfortable Dateneingabe auf Handheld nicht möglich	⇒	Eingabe auf Desktop, Handheld hauptsächlich als „Viewer“
Handheld-Speicher nicht wirklich persistent	⇒	Desktop als Backup-Medium
Handheld kann unterwegs nicht an Internet oder Firmen-Netzwerk angeschlossen werden	⇒	Herunterladen von Informationen bei Synchronisation

Hardware

- Seriell / Modem
 - + einfach zu bedienen, wenig stör anfällig
 - langsam, Kabel nötig, stromfressend
- Infrarot
 - + keine Zusatzhardware
 - stör anfällige Datenübertragung, PCs meist nicht IrDA-fähig
- USB (nur Visor Handspring)
 - + schnellste Übertragungsrates
 - fehlende Kompatibilität zu Linux und Windows 95
- (IP-)Netzwerk
 - + sinnvoll bei Client-/Server-Anwendungen (z.B. Datenbank)
 - HotSync-Server notwendig, Übertragung unverschlüsselt

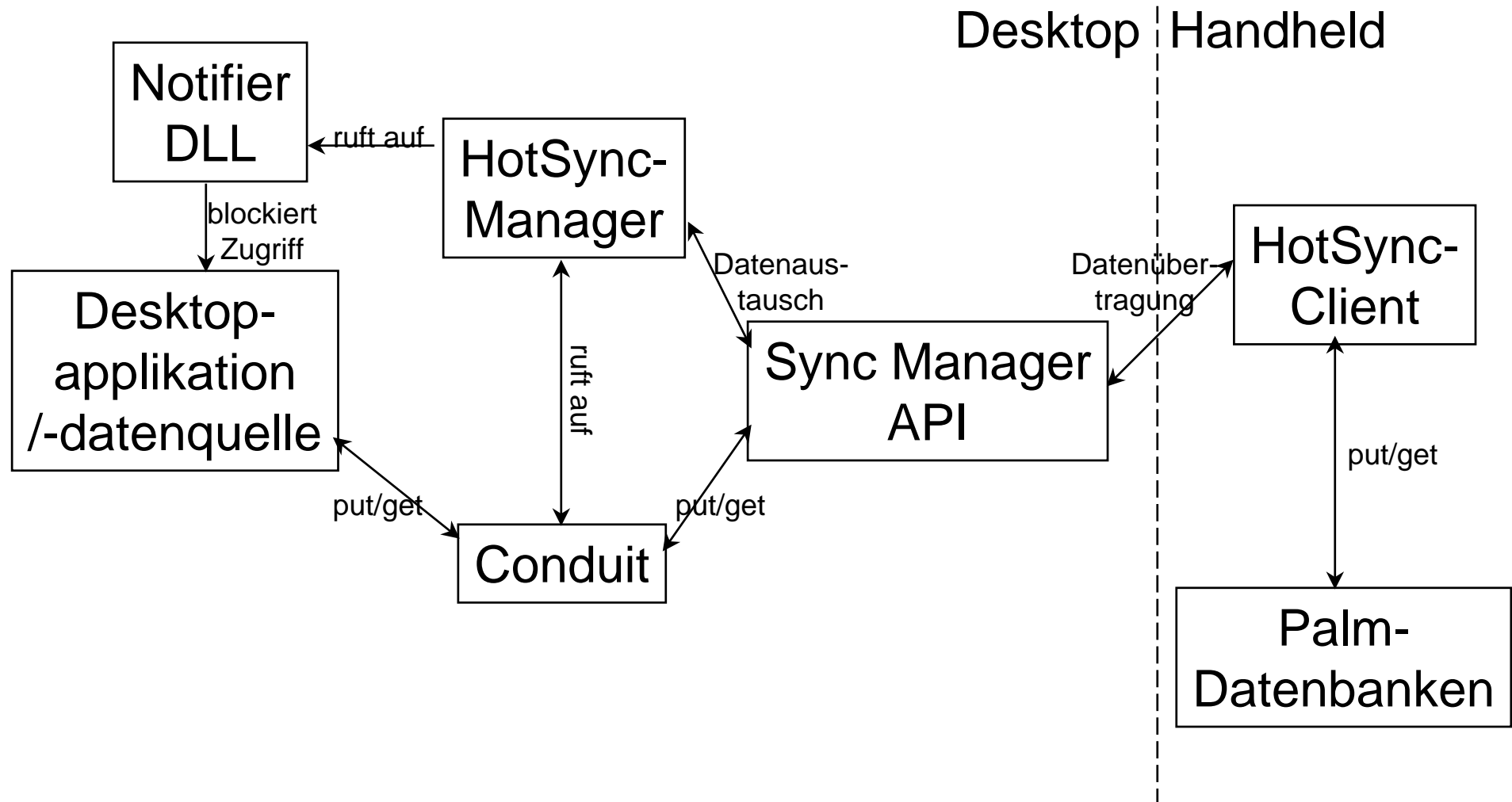
Was ist ein Conduit?

- Bibliothek
- dient zur Synchronisation zwischen Desktop und Handheld
- läuft auf Desktop, nicht auf Handheld
- wird vom HotSync-Manager während HotSync aufgerufen

Was macht ein Conduit?

- wird vom HotSync Manager während des HotSync-Prozesses aufgerufen
- öffnet und schließt Datenbanken auf dem Palm
- stellt fest, in welche Richtung Daten kopiert werden sollen (auf Palm, auf Desktop, beide Richtungen)
- Records auf Desktop und/oder Palm hinzufügen, löschen, ändern
- Datenbanken konvertieren (Haldheld Format, Desktop-Datei-Format)
- keine Interaktion mit dem Benutzer
- protokolliert Aktionen und Fehler in Logdatei

Synchronisation Überblick



Wie läuft der HotSync ab?

- 1. Benutzererkennung und Auffinden des Desktop-Pfades**
eindeutige UserID
jeder Benutzer hat ein Verzeichnis
- 2. Feststellen der Art der Synchronisation**
PCID zur Identifikation des PCs
FastSync oder SlowSync?
- 3. Benachrichtigung an Desktop Applikationen**
Notifier DLL benachrichtigt Desktop Applikation
- 4. Erstellen der Conduit-Liste**
- 5. Installation von Datenbanken**
- 6. Ausführen der Conduits**

Wie läuft der HotSync ab?

7. Zweite Installation

für Datenbanken, die von Conduits erstellt wurden

8. Backup von Datenbanken

9. Update von Synchronisations-Informationen

PCID, Sync Time, Kurzprotokoll

10. Benachrichtigung an Desktop Applikationen

Unlocking

11. Benachrichtigung an Handheld Applikationen

Initialisierung bzw. Verarbeitung der Änderungen

Welche Arten von Conduit gibt es?

- **Backup**

keine Desktop-Komponente, nur Backup
(z.B. Highscore eines Palm-Spieles)

- **nur Up- oder Download**

Conduit kopiert nur Daten von Palm auf Desktop
oder umgekehrt (z.B. Dokumentenbetrachter)

- **Zwei-Wege-Synchronisation**

Conduit gleicht Daten auf Palm und Desktop in
beiden Richtungen ab (z.B. Adressbuch)

- **transaktionsbasiert**

Conduit überträgt z.B. Eingaben von Handheld an
Desktop, wo sie dann verarbeitet werden.

Wähle einfachste Art, die die Anforderungen erfüllt!

Minimal-Funktionalität eines Conduits

- Ein Mechanismus zum Installieren und De-installieren

CDK <3.0: Registry-Einträge direkt hinzufügen und entfernen

CDK ≥3.0: API-Funktionen

- 4 Funktionen:

- **GetConduitName**

liefert den Namen des Conduits zurück

- **GetConduitVersion**

liefert die Version des Conduits zurück

- **OpenConduit**

eigentliche Synchronisations-Funktionalität

- **GetConduitInfo**

liefert zusätzliche Informationen wie Standard-Sync

- optional: **ConfigureConduit**

Windows-Dialog zur Konfiguration des Conduits

- **Log-Einträge**

besonders wichtig, weil HotSync ohne Interaktion mit User erfolgen soll

Vier Wege zum Conduit

Backup-Conduit nutzen

- falls Desktop nicht auf Daten zugreifen muss
- kein Entwicklungsaufwand

Sync Manager API

- “low-level”-Zugriff auf Palm-Datenbanken
- Logik muss komplett implementiert werden

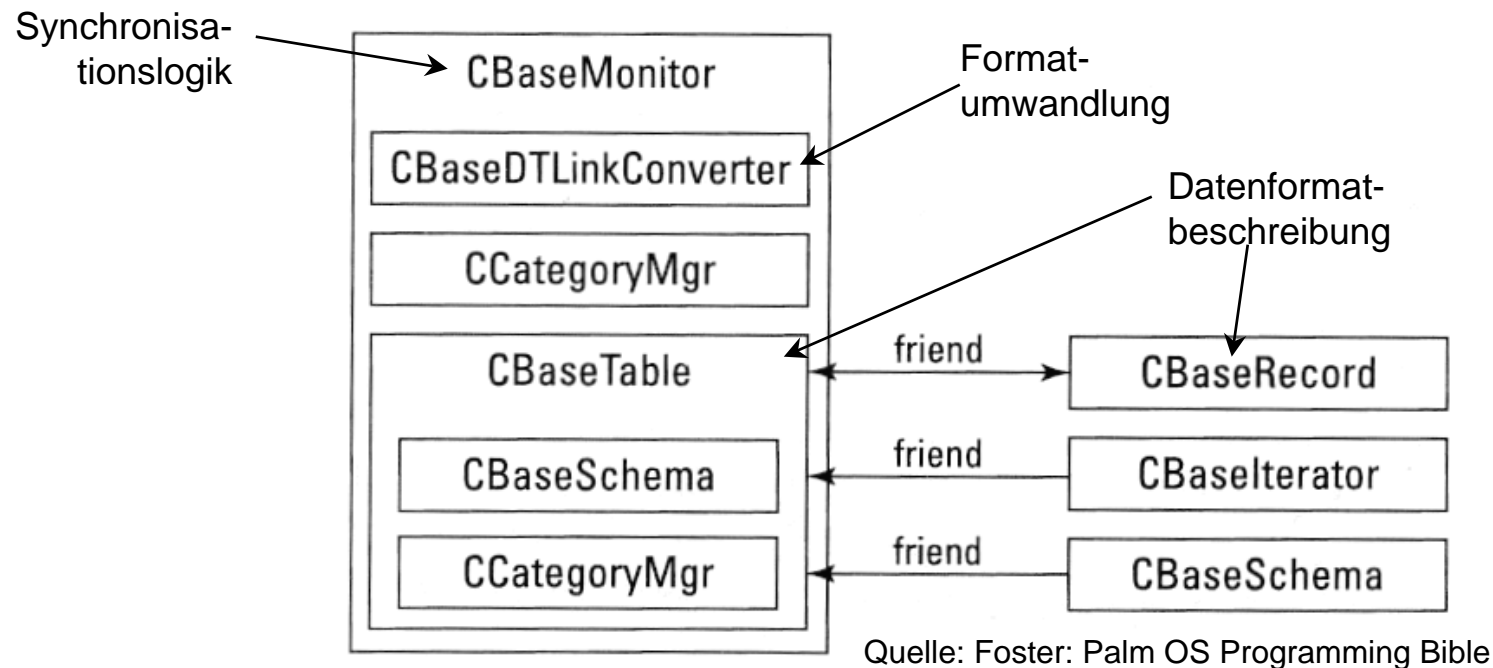
MFC Base Classes

- geben komplette Conduit-Funktionalität vor
- Überschreiben der Methoden, die spezifisch für Programm sind (z.B. Dateiformat)
- nur für Windows verfügbar
- Quellcode nicht offen

Generic Conduit Base Classes

- nicht von Palm
- Open Source

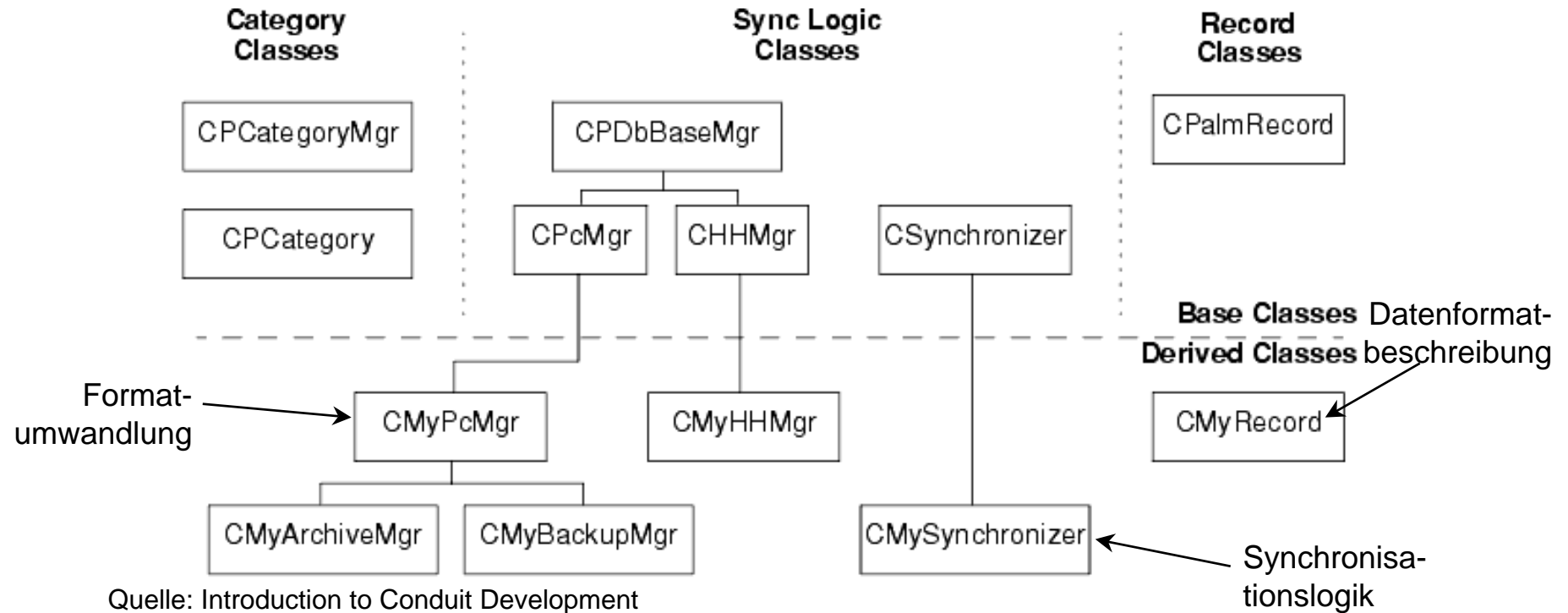
Palm MFC Base Classes



Vorgehen:

- Ableiten der Base Classes
- Überschreiben einzelner Methoden durch eigene

Generic Conduit Base Classes



Vorgehen:

- Ableiten der Base Classes
- Überschreiben einzelner Methoden durch eigene

Sync Manager API

Die Sync Manager API umfasst folgende Arten von Methoden:

- Registrierung des Conduits
- Öffnen von Datenbanken
- Lesen und Schreiben von Records
- Bearbeiten von Kategorien
- Schließen von Datenbanken
- De-installation des Conduits

 Gesamte Semantik muss “von Hand”
implementiert werden

Zusammenfassung (Synchronisation)

- Ein Conduit ist ein Software-Modul, das vom HotSync-Manager aufgerufen wird, um Daten zwischen Handheld und PC zu synchronisieren
- HotSync erfordert keine Benutzer-Interaktion und schreibt alle Meldungen in eine Log-Datei.
- Die vier möglichen Arten von Synchronisation sind Backup, nur Up- oder Download, Zwei-Wege-Synchronisation und transaktionsbasiert
- Es gibt drei Wege, um ein Conduit zu entwickeln: Palm MFC Base Classes, Palm Generic Conduit Base Classes oder direkte Aufrufe der Sync Manager API
- Für ein reines Backup der Daten reicht das Backup-Conduit

Literatur

- Lonnon R. Foster: Palm OS Programming Bible, 2000
- Neil Rhodes, Julie McKeehan: Palm Programming - The Developer's Guide, 1998
- Palm File Format Specification
<http://www.palmos.com/dev/tech/docs/FileFormats/FileFormatsTOC.html>
- Introduction to Conduit Development
<http://www.palmos.com/dev/tech/docs/conduits/win/IntroToConduitsTOC.html>

(Die Links werden auch online zur Verfügung gestellt.)

**Viele Dank für Eure
Aufmerksamkeit!**