Willkommen bei POI

by Andrew C. Oliver, Glen Stampoultzis, Jens Lorenz, Rainer Klute

1. Nachrichten

1.1. Übersetzungen

Das POI-Übersetzungsprojekt hat begonnen. Den Anfang machen spanisch, japanisch und deutsch. Andere Sprachen sind herzlich willkommen. Machen Sie mit!

1.2. Logo-Wettbewerb

Die Wahl für das POI-Logo ist beendet. Danke für Ihre Stimmen.

2. Zweck

Das POI-Projekt besteht aus Java-APIs zum Erstellen und Bearbeiten von Dateiformaten, die auf dem Microsoft-Dateiformat »OLE-2 Compound Document« beruhen. Dateien in diesem Format sind unter anderem die meisten Microsoft-Office-Dateien, wie zum Beispiel Excel- und Word-Dateien.

Grundsätzlich versuchen wir, möglichst viel mit anderen Projekten zusammenzuarbeiten, um die gewünschten Funktionalitäten zur Verfügung zu stellen. Einige Beispiele: Für Cocoon werden bald Generatoren und Serializer zur Verfügung stehen. Wir arbeiten mit Open Office.org zusammen, um das Excel-Dateiformat zu dokumentieren. Für Lucene werden bald Filtermodule zur Verfügung stehen. Wir stellen anderen Projekten Teile des POI-Projektes zur Verfügung, damit diese die POI-Funktionalitäten nutzen können.

2.1. Warum und wann sollte man POI nutzen?

Wir werden diese Frage komponentenweise beantworten. POI besteht aus einer Reihe von Komponenten, die jeweils unterschiedliche Probleme angehen. Das Kürzel »POI« steht für das gesamte Projekt.

Mit **POIFS** können Sie Dateien oder Dokumente, die im OLE 2 Compound Document Format geschrieben wurden, mit Java einlesen. Solche Dateien werden üblicherweise mit der

Copyright (c) @year@ The Apache Software Foundation All rights reserved. $Revision: 1.9 $ $Date: 2004/04/09 13:05:10 $

Copyright © 2002-2005 The Apache Software Foundation. All rights reserved.

MFC-Klassenbibliothek erzeugt. Außerdem können sie POIFS nutzen, um Dateien im OLE 2 Compound Document Format zu schreiben. Damit können sie zum Beispiel den Datenaustausch mit der Windows-Plattform sicherstellen. Wir können guten Gewissens behaupten, daß POIFS die vollständigste Implementierung dieses Dateiformates ist.

Mit **HSSF** können sie Excel-Dateien in Java lesen und schreiben. Sie können auch Excel-Tabellen lesen und modifizieren. Allerdings ist die Schreibfunktionalität im Moment am ausgereiftesten.

2.2. Wofür steht POI ?

POI bedeutet »Poor Obfuscation Implementation« (Schlechte, verschleiernde Implementierung). Warum geben wir unserem Projekt einen so abschätzigen Namen? Nun, das Microsoft OLE 2 Compound Document Format ist einfach schlecht durchdacht. Von seiner Grundidee her ist es ein Dateiarchiv mit einer Struktur, die dem alten DOS-FAT-Dateisystem ähnelt. Die Redmonder haben kein bereits vorhandenes Archivformat wie tar, gzip, zip oder arc genutzt, sondern stattdessen ein eigenes Archivformat erfunden, das keinerlei Standardverschlüsselung oder -komprimierung bietet, das schlecht erweiterbar ist, und das zur Fragmentierung neigt.

POI ist außerdem eine Spezialität der hawaiianischen Küche. Sie wird in Merriam Webster's Dictionary beschrieben als: »Ein hawaiianisches Gericht aus Taro-Wurzeln, die durch Stampfen, Kochen und Kneten zu einer Paste geformt und oft noch ein wenig gegoren wird.« Dies ist witzigerweise eine treffende Beschreibung des Dateiformats.

POI ist also eine Abkürzung. Wenn Sie Abkürzungen nicht mögen, dann denken sie einfach bei POI an das hawaiianischen Gericht. Je nachdem, ob Sie Abkürzungen mögen oder nicht, nutzen sie einfach POI oder Poi, wenn sie dieses Projekt meinen.

3. Komponenten

3.1. Überblick

POI besteht aus mehreren Komponenten, die jeweils unterschiedliche Aufgaben angehen. Beispielsweise dient die Komponente HSSF dazu, Excel-Dateien zu schreiben und zu lesen. Es folgt eine Liste aller Komponenten des POI-Projektes mit einer sehr kurzen Zusammenfassung ihres Zweckes.

3.2. POIFS (POI Filesystem)

POIFS ist der älteste und stabilste Teil des Projektes. POIFS ist unsere Portierung des OLE 2

Copyright (c) @year@ The Apache Software Foundation All rights reserved. $Revision: 1.9 $ $Date: 2004/04/09 13:05:10 $

Copyright © 2002-2005 The Apache Software Foundation. All rights reserved.

Compound Document Formats in reinem Java. Es unterstützt Lesen und Schreiben. Alle anderen Komponenten basieren auf POIFS. Mehr Informationen gibt es auf der POIFS-Seite.

3.3. HSSF

HSSF ist unsere Portierung des Microsoft Excel 97(-2002) Dateiformats in reinem Java. Es unterstützt Lesen und Schreiben. Mehr Informationen gibt es auf der HSSF-Seite.

3.4. HWPF

HWPF ist unsere Portierung des Microsoft Word 97 Datei-Formats in reinem Java. Es unterstützt Lesen und Schreiben. Mehr Informationen gibt es auf der HWPF-Seite. Diese Komponente ist noch nicht sehr weit fortgeschritten. Wir suchen Entwickler, die mitmachen.

3.5. HPSF

HPSF ist unsere Portierung des OLE 2 Property Formats. Property Sets nehmen die Metadaten eines Dokuments auf, wie Titel, Autor und Datum. Sie lassen sich aber auch für applikationsspezifische Aufgaben nutzen. Mehr Informationen gibt es auf der HPSF-Seite.

4. Mitmachen

Sie möchten bei diesem Projekt mitmachen? Hervorragend! Wir brauchen immer begeisterte, fleißige und talentierte Leute, die uns bei den verschiedenen Aufgaben des Projektes helfen. An erster Stelle stehen Hinweise auf Fehler und Vorschläge für neue Funktionen. Ebenso wichtig ist die Dokumentation.

Egal, ob sie Kritik oder Vorschläge haben, oder ob Sie Beiträge in Form von Code oder Dokumentation liefern möchten, immer werden Sie bei uns ein offenes Ohr finden. Und nicht zuletzt brauchen wir Java-Programmierer, die sich durch die zahlreichen Ecken und Kanten der Microsoft-Dateiformate hindurchwühlen und uns dabei helfen, diese Formate auf die Java-Plattform zu portieren.

Wenn Sie motiviert sind und Zeit haben, tragen Sie sich in unsere Mailing-Listen ein, und machen sie mit! Bei der Einarbeitung helfen wir Ihnen gerne.

Copyright (c) @year@ The Apache Software Foundation All rights reserved. $Revision: 1.9 $ $Date: 2004/04/09 13:05:10 $

Copyright © 2002-2005 The Apache Software Foundation. All rights reserved.