

# Die IBM Rational Migration Extension for System i (RMEi) Eine zukunftsweisende Lösung

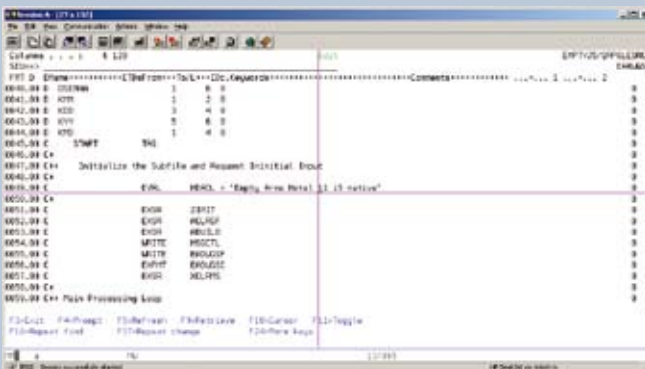
Der Paradigmen-Bruch zwischen traditioneller prozeduraler Programmierung und objektorientierter Welt erregt auch im Krisenjahr 2009 die Gemüter der Entwickler und Software-Architekten. Die Kernfrage, um die es dabei geht: Wie können die Stärken beider Ansätze (schnell und produktiv entwickeln sowie modernste Technologie für den Anwender anbieten) sinnvoll kombiniert werden?

Auf diese Frage gibt es genau eine Antwort: Umsteigen auf die IBM Business-Sprache EGL (Enterprise Generation Language). EGL ist eine Programmiersprache der neuesten Generation, die sowohl den client- als auch den serverseitigen Bedarf bei der Entwicklung von Geschäftsanwendungen

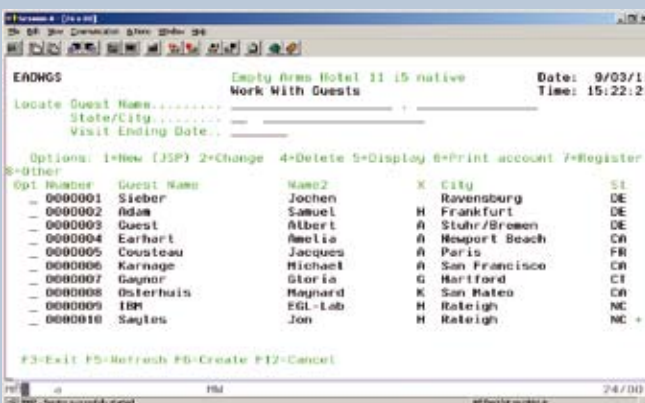
berücksichtigt: Moderne Frontends, basierend auf JSF und Ajax-Technologie, lassen sich ebenso herstellen wie klassische Batch-Programme mit Text-User-Interface à la 5250. Das Schöne daran: Aus EGL heraus kann man direkt Java-Programme erzeugen, aber weiterhin auch native i-Programme. Doch ganz

besonders wichtig: EGL ist keine objektorientierte, sondern eine prozedurale Programmiersprache, in der die modernen SOA-Aspekte implizit eingebunden sind. Die bisher fehlende tragfähige Brücke zwischen traditioneller und objektorientierter Welt ist somit gebaut. Die EGL Entwicklungsumgebung ist in

## 1 VERFOLGUNG DER ÄNDERUNGEN



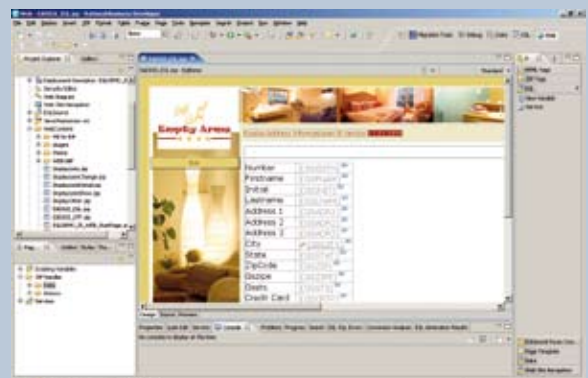
## 2 5250-ANSICHT



## 3 RMEI-ANSICHT

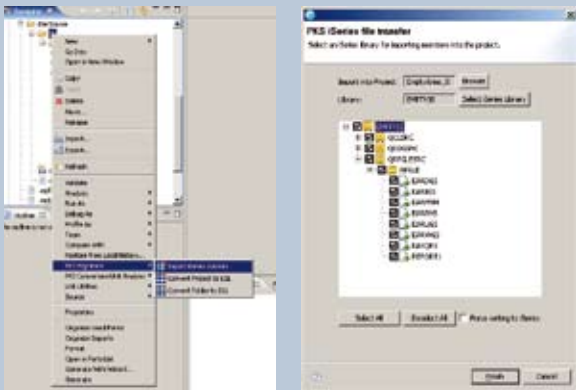


## 4 BILDSCHIRMGESTALTUNG IN EGL

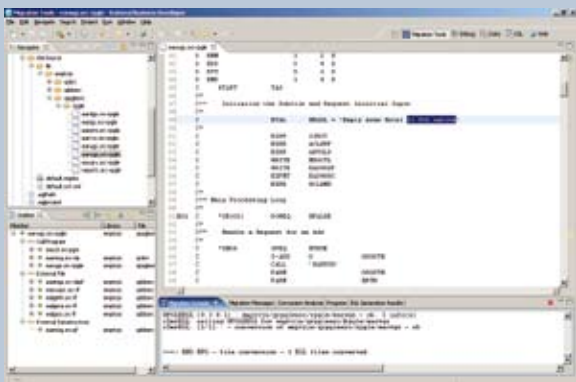




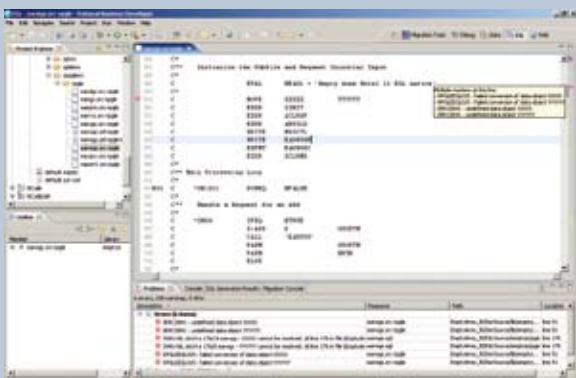
## 6 IMPORT WIZARD



## 7 MIGRATION



## 8 PROBLEMANSICHT BEI FEHLERN



alle verwendeten PF, LF, DSPF und PRTF Sourcen inklusive Referenzfiles.

Über einen Import Wizard werden die entsprechenden Sourcen aus den System i Members in eine geeignete EGL Projektstruktur übertragen (vgl. Abb. 6).

Alle Sourcen werden in eine standardisierte Projektstruktur übertragen. Hierbei wird für jede Library ein eigenes Verzeichnis angelegt und entsprechende Unterverzeichnisse für Sourcefiles und Sourcemembers inklusive der verschiedenen Sourcearten.

Nun erfolgt ein Wechsel der Perspektive auf die „RPG – EGL Migration“. In dieser Perspektive finden sich alle für die Migration benötigten Werkzeuge.

2. Die Migration kann nun gestartet werden. Sie kann erfolgen Source für Source oder als Gesamtmigration für ein komplettes Projekt- bzw. Teilprojekt.

Während die Migration läuft, werden auf der „Migration Tooling Console“ alle ausgeführten Schritte angezeigt. Nach Abschluss der Migration erfolgt auf der Konsole eine Meldung über die Anzahl der migrierten Programme.

Hier wird nun die am Anfang des Artikel geänderte RPG Source nochmals geändert in „11 EGL native“ (Abb. 7).

Um zu sehen, was passiert, wenn in der Migration der Programme ein Fehler auftritt, simulieren wir ein Problem mit einem MOVE Befehl im Programm.

Der Befehl MOVE XXXX YYYYYY arbeitet mit nicht deklarierten Feldern XXXX und YYYYYY (Abb. 8).

**Karlheinz Peter ■**

So leicht und flüssig können also vorhandene RPG-Anwendungen mit der RMEi Web-API in ein Browser-Frontend überführt und nahtlos mit Neu-Entwicklungen in JSP-Technologie eingebunden werden.

Doch nun zur zweiten wichtigen Option der RMEi: dem Konverter zur Migration von RPG nach EGL. Dazu

beschreiben wir nachfolgend die einzelnen Schritte, die nötig sind, um existierende RPG-Anwendungen nach EGL zu überführen:

1. Zunächst werden die existierenden Sourcen von der iSeries in die EGL Rational Entwicklungsumgebung überführt. Hierbei werden nicht nur die RPGLE Sourcen benötigt, sondern auch

**CLICK TO**

[www.midrange.de](http://www.midrange.de)

Den vollständigen Artikel finden Sie in unserer aktuellen Online-Ausgabe.