

IBM DB2 Web Query 2.1 – Data Migrator Teil 7

Seit März 2015 ist das neueste Mitglied der DB2 Web Query-Familie verfügbar: **IBM DB2 Web Query for i DataMigrator ETL Extension**. In den ersten Artikeln habe ich Ihnen eine grobe Übersicht über die Möglichkeiten von DataMigrator gegeben. Dies wurde dann in Form eines einfachen Datenflusses exemplarisch gezeigt. In diesem Artikel werden weitere und komplexere Funktionalitäten von DataMigrator und insbesondere von Datenflüssen dargestellt.

Weitere Funktionen von Datenflüssen

Im letzten Artikel habe ich Ihnen die Hauptkomponenten von DataMigrator gezeigt. Dazu habe ich einen einfachen Datenfluss erzeugt, in dem die Spalten einer Tabelle eins-zu-eins an die Zieltabelle weitergereicht wurden. Im Folgenden verwende ich diesen Datenfluss als Ausgangspunkt für die Erläuterung von weiteren und komplexen Funktionalitäten. Dies erfolgt in mehreren Schritten:

- Vorbereitende Maßnahmen (wurde bereits in vorhergehenden Artikeln beschrieben)
- Einen Fluss kopieren (wurde bereits in vorhergehenden Artikeln beschrieben)
- Hinzufügen von Joins (wurde bereits in vorhergehenden Artikeln beschrieben)
- Hinzufügen von Transformationen (wurde bereits in vorhergehenden Artikeln beschrieben)
- Spalten und Transformationen auswählen (wurde bereits in vorhergehenden Artikeln beschr.)
- Datenziele hinzufügen (wurde im letzten Artikel beschrieben)
- Eine Validierung hinzufügen (wurde im letzten Artikel beschrieben)
- **Ein zweites Datenziel hinzufügen** (wird in diesem Artikel beschrieben)
- **Fluss ausführen** (wird in diesem Artikel beschrieben)

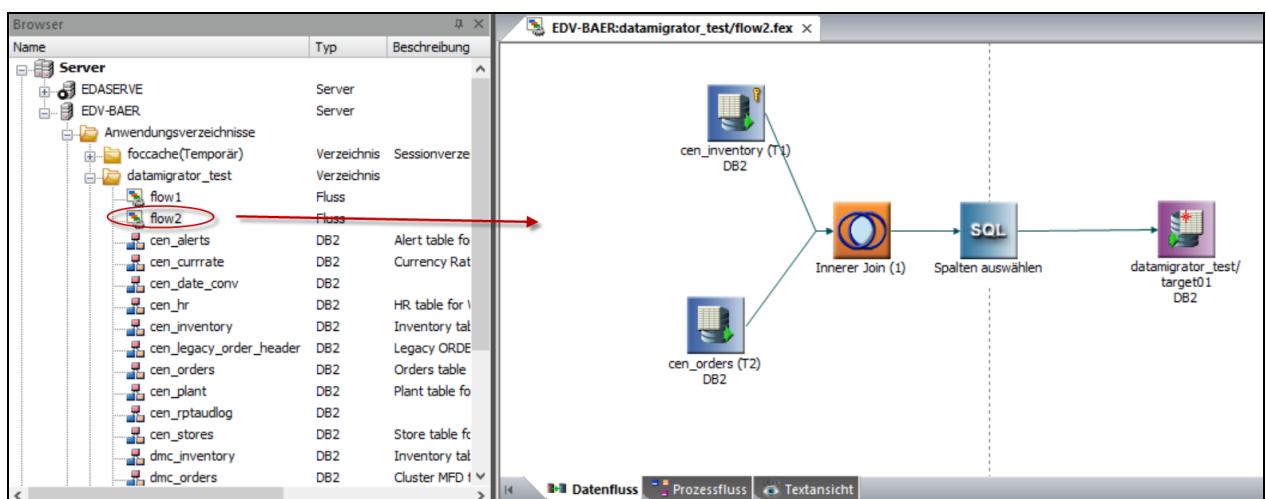


Abbildung 1: Darstellung des Flusses flow2

Wie bei flow1 besteht die Datenquelle aus der Tabelle inventory und dem Datenziel target01. Im letzten Artikel haben wir dann eine zweite Datenquelle *cen_orders* hinzugefügt. Beide Tabellen wurden durch einen Inner Join miteinander verbunden. Dieser Datenfluss wird nun weiter bearbeitet.

Ein zweites Datenziel hinzufügen

Mit DataMigrator ist es möglich, mehrere Datenziele in einem Datenfluss zu definieren. Dazu müssen wir ein neues Zielobjekt in unserem Datenfluss definieren. Da dieses Datenziel noch nicht existiert, müssen wir es mit Hilfe der Spalten, die im Objekt *SQL Spalten auswählen* definiert sind.

Unser Datenfluss wird die Basistabelle zwar erstellen, diese kann aber auch durch andere Datenflüsse verändert werden.

Die Erstellung eines neuen Datenziels erfolgt mit:

Kontextmenü Arbeitsbereich rechts vom SQL-Objekt → Ziel hinzufügen → Neu

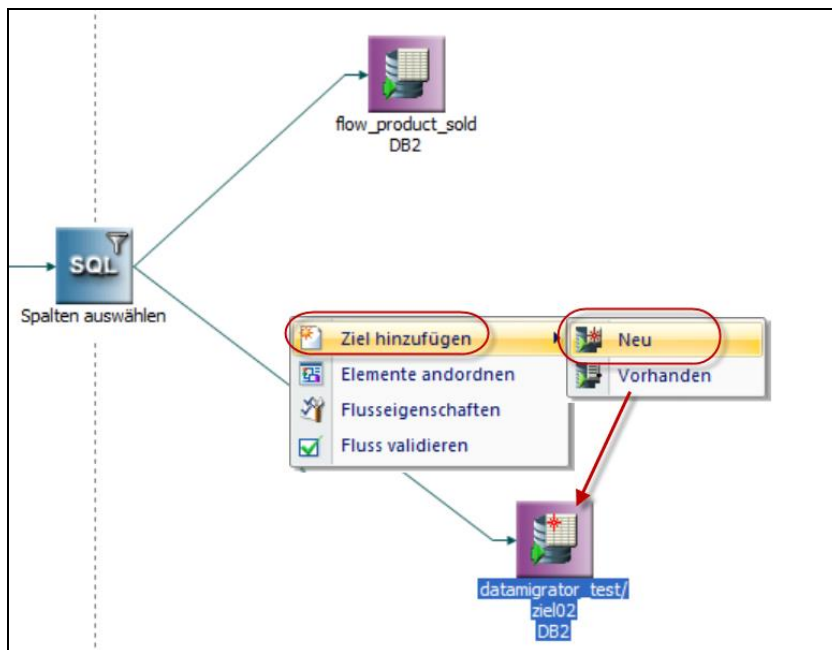


Abbildung 2: Ein neues Datenziel hinzufügen

Die weitere Definition des neuen Datenziels erfolgt über die zugehörigen Eigenschaften (*Kontextmenü → Eigenschaften*). Einige der Eigenschaften sind bereits durch die auch von uns gewünschten Standardwerte vorgelegt.

- Der Adapter ist DB2 (DB2 für i).
- Die Verbindung ist *LOCAL. Dies bedeutet, dass die Ergebnis-Tabelle auf das gleiche System gespeichert werden soll, auf dem DataMigrator läuft.

Die folgenden Parameter habe ich wie folgt angepasst:

- **Synonym** : datamigrator_test/flow_new_prod
Das Synonym für die neu zu erstellende Tabelle *flow_new_prod* wird im Ordner *datamigrator_test* gespeichert.
- **Tabelle** : qwqadmtest/flow_new_prod
Die Tabelle *flow_new_prod* wird in der Bibliothek QWQADMTEST erstellt.
- **Ladetyp** : Datensätze vom Speicher einfügen

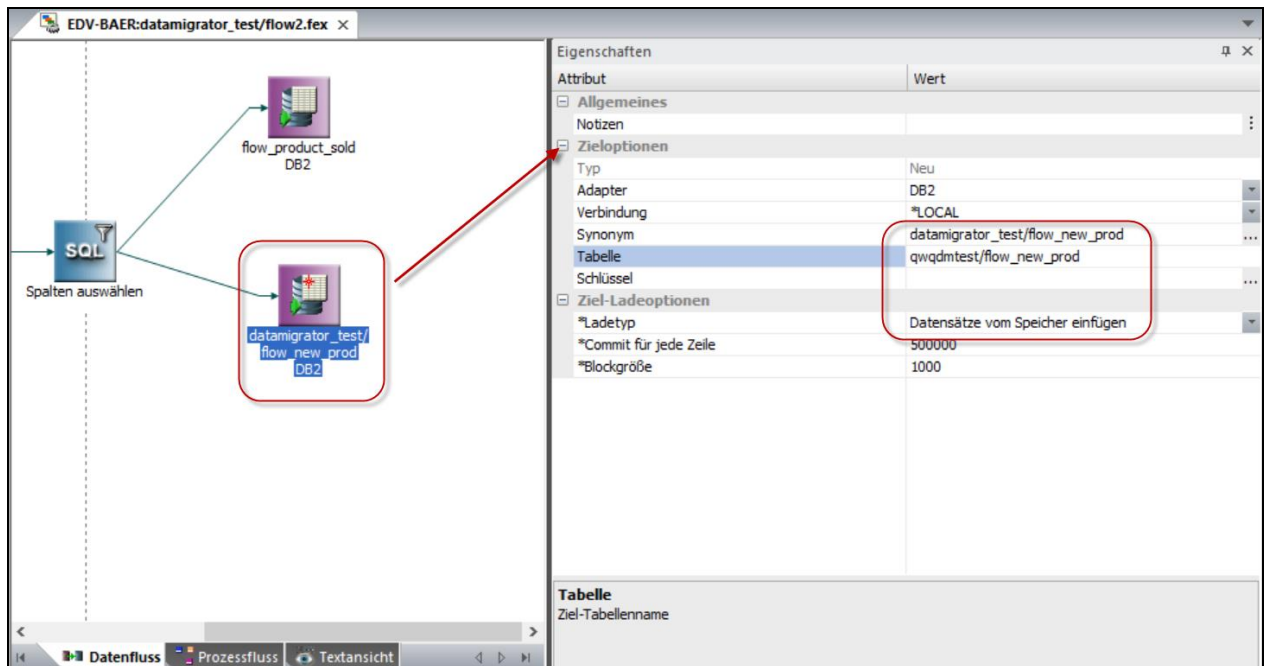


Abbildung 3: Eigenschaften des neuen Datenziels

Hinweis:

Die Eigenschaftsoptionen, die mit einem Stern (*) gekennzeichnet sind, gelten für den gesamten Fluss. Dies bedeutet auch, dass diese Optionen für alle Datenziele gleichartig gesetzt werden müssen. Da wir einen Fluss mit zwei Datenzielen haben, müssen wir sicherstellen, dass die mit einem Stern versehenen Optionen für beide Datenziele identisch sind. Ist das nicht der Fall, erhalten Sie eine Warnung, wenn Sie versuchen, den Fluss zu sichern oder auszuführen.

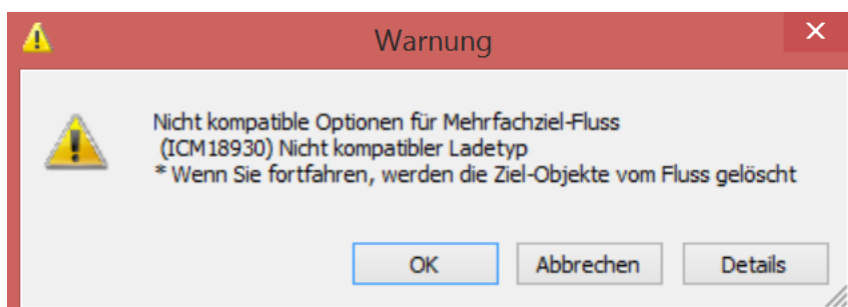


Abbildung 4: Warnungshinweis, falls eine Stern-Option nicht für alle Datenziele kompatibel ist

Hinweis;

Bei meinem Test habe ich für das neue Datenziel den Ladetyp geändert. Danach habe ich versucht, den Fluß zu sichern und habe den Hinweis wie in der vorigen Abbildung erhalten. Aber was dann passierte, hat mich doch etwas geschockt: **Beider Datenziele waren verschwunden**. Ich habe dazu sofort eine Info an das Labor in Rochester geschickt.

Aber: wenn man die Warnung genauer liest, erkennt man, dass auf dieses Phänomen hingewiesen wird:

***Wenn Sie fortfahren, werden die Ziel-Objekte vom Fluss gelöscht.**

Abhilfe: Das erste Datenziel kann einfach nochmals hinzugefügt werden mit der Auswahl „Bereits vorhanden“. Dazu wird dann einfach das Synonym *slow_product_sold* verwendet. Das zweite Datenziel muß nochmals neu hinzugefügt wie oben gezeigt. Also Vorsicht!

In der zu erstellenden Tabelle sind die Spalten PRODUCTNAME, PRODUCTTYPE, PRODUCTNUMBER, STORECODE und SHIPDATE Schlüsselfelder. Das heißt, dass jede Ergebniszeile eine eindeutige Kombination dieser Spalten enthält. Wenn diese Spalten als Schlüsselspalten gekennzeichnet werden, wendet DataMigrator dies auf die zugrunde liegende Tabelle auch bei nachfolgenden Datenänderungen an. Zusätzlich wird dadurch auch die Datenintegrität sichergestellt.

Für die Realisierung doppelklicken Sie bitte auf das neue Datenziel und kennzeichnen die entsprechenden Spalten als Schlüssel (siehe folgende Abbildung).

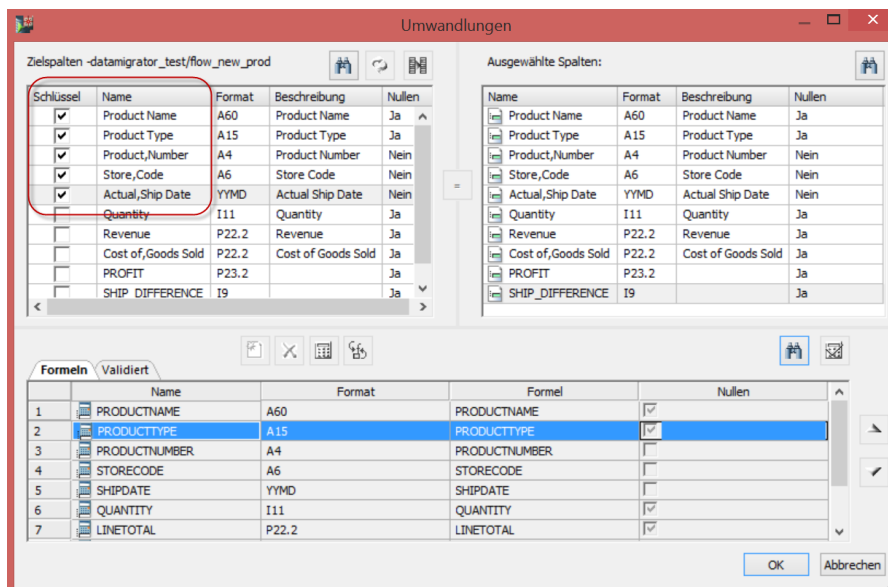


Abbildung 5: Schlüsselfelder kennzeichnen

Unser Datenfluss mit den zwei Datenquellen und den zwei Datenzielen ist nun fertig und sieht wie folgt aus:

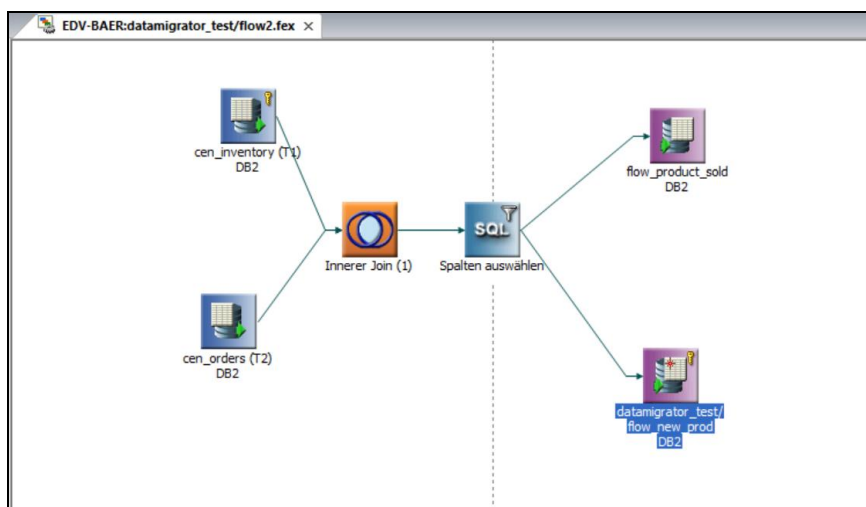


Abbildung 6: Datenfluss flow2 mit zwei Datenquellen und zwei Datenzielen

Jetzt können Sie Ihren Fluss auch bedenkenlos sichern!

Fluss ausführen

Unser Datenfluss ist nun fertiggestellt. Die beiden Datenziele *flow_new_prod* und *flow_product_sold* werden beim Ausführen des Flusses mit den Daten geladen, die bei den Datenfluss-Spezifikationen angegeben wurden.

Lassen wir also den Fluss laufen. Dazu gibt es wiederum verschiedene Möglichkeiten:

- mit Hilfe des Kontextmenüs von *flow2* im Navigationsbaum und Auswahl *Ausführen*

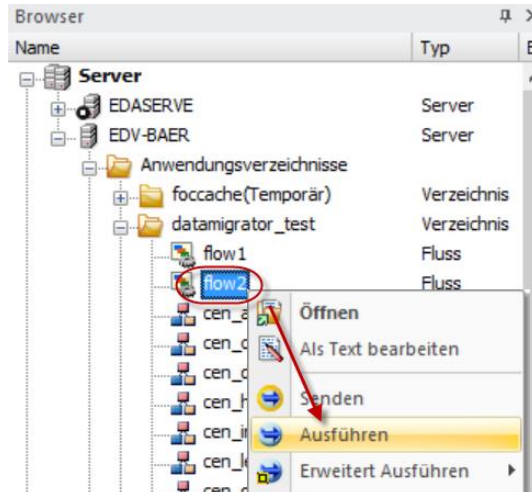


Abbildung 7 Fluss über das Kontextmenü ausführen

- Falls der Fluss geöffnet ist, kann die Ausführung über die Auswahl *Ausführen* über das zum Menü *Fluss* gehörige Band gestartet werden.

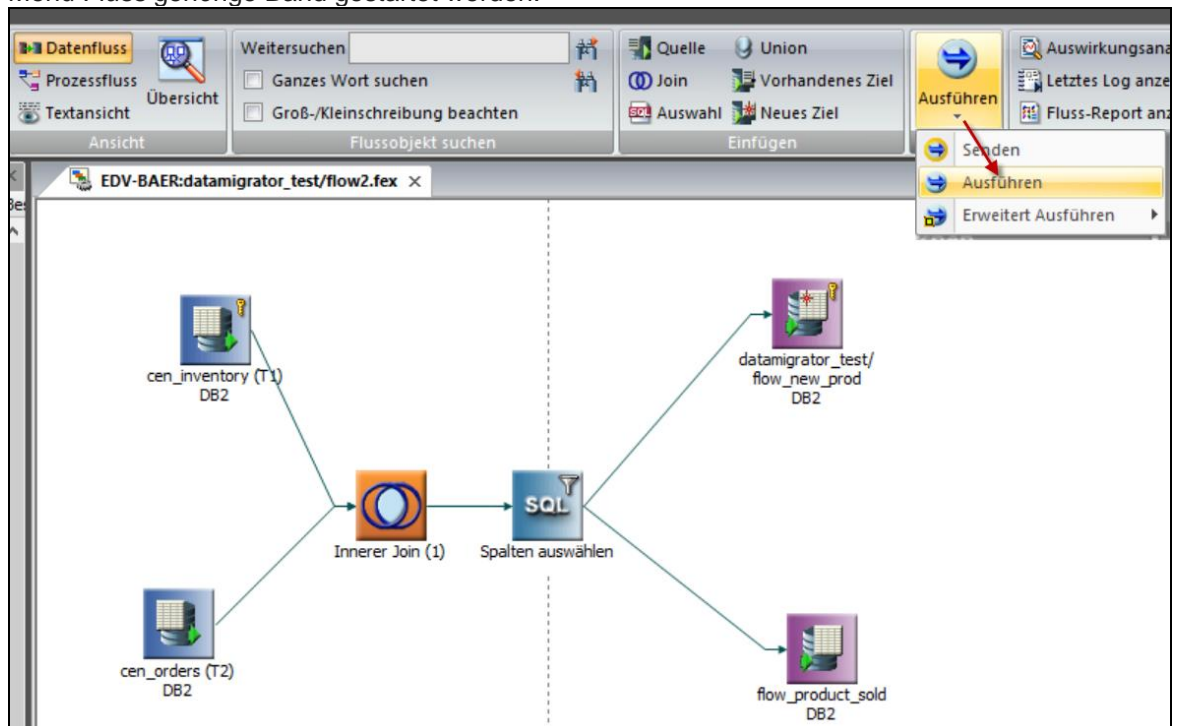


Abbildung 8: Fluss über das Menüband ausführen

Sobald die Ausführung beendet ist, kann das Konsolen-Protokoll als neuer Tab des DMC-Fensters angezeigt werden.

Wenn ein Fluss mit *Senden* und nicht mit *Ausführen* gestartet wird, wird der Fluss als Hintergrund-Prozess ausgeführt. Folglich zeigt der Konsolen-Log nur, dass die Anforderung erstellt und erfolgreich ausgeführt wurde.

Um weitere Informationen zu dem Prozessverlauf zu erhalten, muss der Prozess daher mit *Ausführen* gestartet werden. Der Log ist auch zur Analyse von Problemen wichtig, die beim Ausführen des Flusses aufgetreten sein können.

Hinweis:

Zum Testen eines Flusses während der Erstellung sollte dieser immer mit *Ausführen* und nicht mit *Senden* ausgeführt werden. Dadurch werden eventuell auftretende Probleme im Konsolen-Log aufgezeichnet. Sobald ein Fluss fertiggestellt und ausgetestet ist, kann die Ausführung mit der Auswahl *Senden* im Hintergrund erfolgen.

Zur Anzeige des Logs wählen Sie im Menüband *Letztes Log anzeigen*.

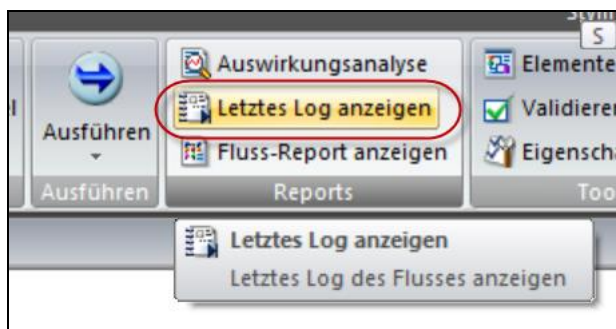


Abbildung 9: Auswahl zur Anzeige des Logs

The image shows a screenshot of the IBM Data Migrator console. The 'Log Messages' tab is active, displaying a list of log entries. The table has columns for 'Name', 'Type', 'Datetime', 'Message Code', 'Log Messages', 'Application', 'Name', and 'Job ID'. The log entries show the progress of a flow named 'flow2', including commit messages and a final 'Finished' status. The 'Return Code = 0' and 'Finished' messages are circled in red.

Name	Type	Datetime	Message Code	Log Messages	Application	Name	Job ID
datamigrator_test	flow2	2015/02/17 20:00:08	(ICM18745)	REFERENCE...AT TRANS 24000	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:08	(ICM18745)	Commit forced at: 24369 for 600 row(s)	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:09	(ICM18745)	Commit forced at: 24964 for 595 row(s)	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:09	(ICM18745)	REFERENCE...AT TRANS 25000	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:09	(ICM18745)	Commit forced at: 25555 for 591 row(s)	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:09	(ICM18745)	REFERENCE...AT TRANS 26000	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:09	(ICM18745)	Commit forced at: 26149 for 594 row(s)	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:09	(ICM18745)	Commit forced at: 26741 for 592 row(s)	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:09	(ICM18745)	REFERENCE...AT TRANS 27000	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:09	(ICM18745)	Commit forced at: 27335 for 594 row(s)	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:09	(ICM18745)	Commit forced at: 27930 for 595 row(s)	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:09	(ICM18745)	REFERENCE...AT TRANS 28000	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:09	(ICM18745)	Commit forced at: 28535 for 605 row(s)	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:09	(ICM18745)	REFERENCE...AT TRANS 29000	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:10	(ICM18745)	Commit forced at: 29090 for 90 row(s)	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:10	(ICM18744)	Ending Load	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:10	(ICM18040)	Return Code = 0	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:10	(ICM18076)	Request: datamigrator_test/flow2 - finished processing	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:10	(ICM18007)	CPU Time : 3129	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:10	(ICM18031)	Finished	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:10	(ICM18072)	Elapsed run time 0:00:09	datamigrator_test	flow2	20150217200000_4649dcb
datamigrator_test	flow2	2015/02/17 20:00:10	(ICM18975)	End of Log Record for datamigrator_test/flow2	datamigrator_test	flow2	20150217200000_4649dcb

Abbildung 10: Eine Seite des Ausführungsprotokolls (Log)

Die im Log dargestellten Informationen sind auch abhängig von der Arbeitsumgebung und können daher von den auf Ihrem System erzeugten abweichen. Wichtig sind die beiden Informationen *Return Code = 0* und *Finished*, womit ein erfolgreicher Ablauf dokumentiert wird.

Beachten Sie auch, dass sich die Optionen des Menübands der Auswahl des Logs spezifisch angepasst haben.

Vorschau:

In den nächsten Artikeln werden die weiteren Funktionalitäten des DataMigrator for i vorgestellt und näher beleuchtet.

Bis dahin wünsche ich Ihnen weiterhin viel Spaß beim Vermehren Ihrer Fertigkeiten.

Den Autor Theo Bär erreichen Sie unter EDV-Beratung Theo Bär - Ringmauerweg 1 - 69250 Schönau -
Tel. (+49) 6228 912 630 - e-Mail info@edv-baer.com