

IBM DB2 Web Query 2.1 – Data Migrator Teil 8

Seit März 2015 ist das neueste Mitglied der DB2 Web Query-Familie verfügbar: *IBM DB2 Web Query for i DataMigrator ETL Extension*. In meinen bisherigen Artikeln zu diesem Thema habe ich neben den vielfältigen Eigenschaften von DataMigrator die Erstellung von Datenflüssen beschrieben. Im aktuellen Artikel möchte ich Ihnen zeigen, welche Möglichkeiten DataMigrator bei der Erstellung von *Prozessflüssen* bietet.

Übersicht Prozessfluss

Mit einem Prozessfluss wird kontrolliert, auf welche Art und Weise ein oder mehrere Datenflüsse ausgeführt werden. Die Schritte eines Prozesses und deren Reihenfolge werden durch Positionierung von Objekten innerhalb des Arbeitsbereiches einschließlich deren Interaktionen definiert. Ein Prozessfluss besteht aus:

- einem *Startobjekt*, welches den Beginn des Prozessflusses definiert.
- *Datenflussobjekten*, die zeigen, welche Daten aus Datenquellen extrahiert und in Datenziele kopiert werden sollen.
- *E-Mail-Objekten*, die Benutzer über den Status des Prozessverlaufs an bestimmten Punkten informieren sollen.
- *Gespeicherte Prozeduren*, die zahlreiche zusätzliche Aufgaben vor oder nach der Extraktion und in einem Datenfluss definierten Ladeschritten ausführen.
Hinweis:
Diese Gespeicherten Prozeduren sind Bestandteil von DataMigrator for i und nicht der Datenbank.
- *Verbindungsobjekte*, die als Pfeile dargestellt werden. Diese bestimmen die Ausführungslogik für die anderen Objekte, die in dem Prozessfluss enthalten sind. Beispielsweise wird damit bestimmt, was als nächstes ausgeführt werden soll, wenn ein Schritt fehlschlägt oder erfolgreich ausgeführt wird.
- *Gruppenobjekte*, die den Flussverlauf für eine Untermenge von Objekten innerhalb des gesamten Flusses definieren. Beispielsweise sollen zwei Objekte einer Gruppe gleichzeitig und nicht nacheinander ausgeführt werden. Danach soll ein einzelnes Objekt ausgeführt werden. Dies könnte zum Beispiel eine E-Mail-Benachrichtigung sein, die darauf wartet, bis beide Jobs einer Gruppe beendet sind.

Wie bereits erwähnt, wird für jeden Datenfluss automatisch ein Prozessfluss erzeugt. Sobald ein Datenfluss ausgeführt oder zur Ausführung abgeschickt wird, wird der Prozessfluss selbst ausgeführt. Der Prozessfluss indessen steuert den Datenfluss.

Schauen wir uns den Datenfluss *flow2* an, den wir im letzten Artikel erstellt haben.

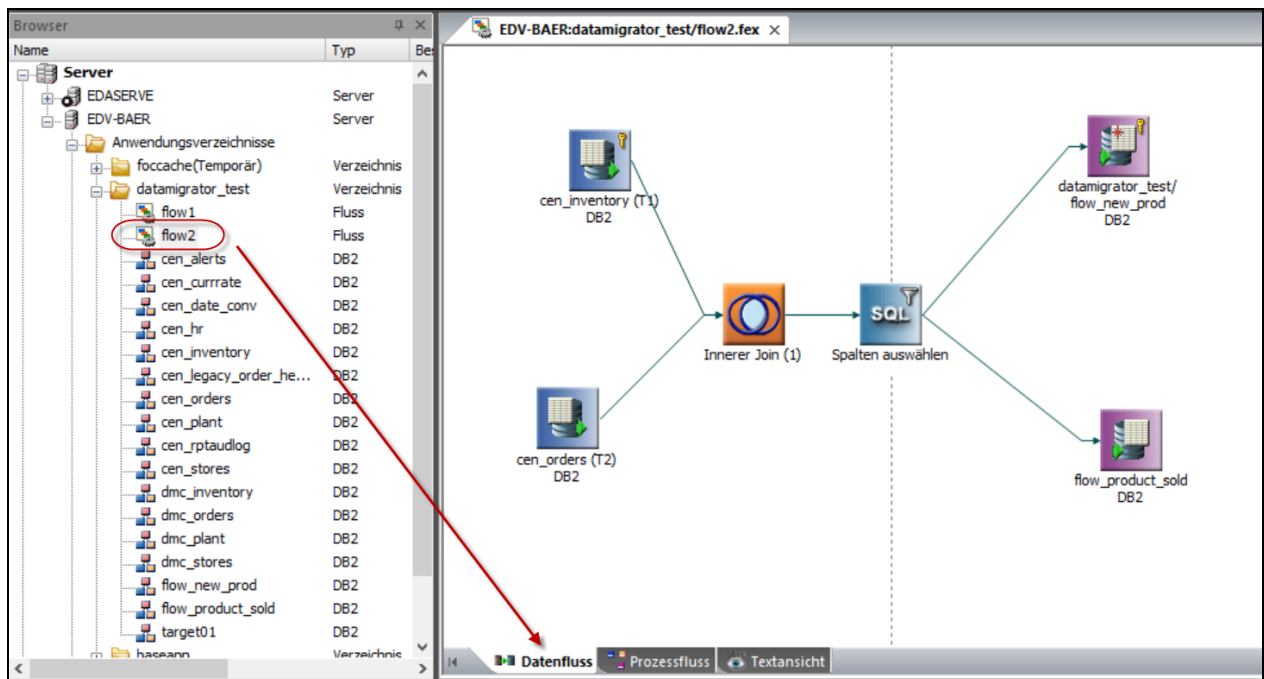


Abbildung 1: Darstellung des Datenflusses flow2

Neben dem in Abbildung 1 aktiven Tab *Datenfluss* befindet sich ein weiterer Tab *Prozessfluss*.

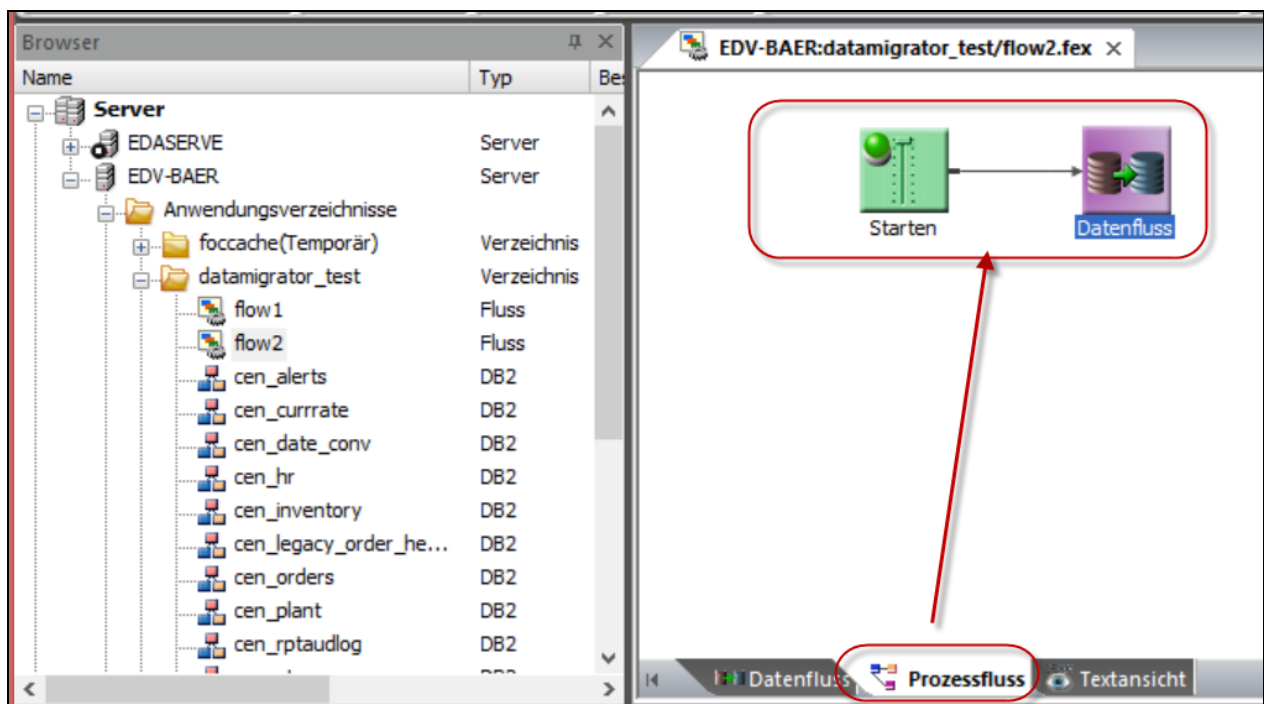


Abbildung 2: Prozessfluss des Datenflusses flow2

Der in Abbildung 2 dargestellte Prozessfluss wurde automatisch erzeugt.

Im einfachsten Fall brauchen Sie überhaupt keinen Prozessfluss erstellen, das macht DataMigrator für Sie. Allerdings gibt es viele Gründe, weitere Schritte dem Prozessfluss hinzuzufügen: Zusammenlegen von mehreren Datenflüssen in einen Prozess, Rückmeldungen per E-Mail erzeugen, Fehlerbehandlungen durchführen und so weiter.

Einen Prozessfluss erstellen

Im folgenden Beispiel erstelle ich einen Prozess, der den Datenfluss *flow2* aus dem letzten Artikel verwendet. Abhängig davon, ob der Datenfluss erfolgreich ausgeführt wurde oder nicht, wird in eine von zwei Prozeduren verzweigt. Die zugehörigen Ergebnisse werden in einer Log-Datei aufgezeichnet, damit eine schnelle Fehlersuche möglich ist oder eine unmittelbare Erfolgsmeldung erstellt werden kann.

Hinweis:

Die Bereiche „Erfolgreiche Ausführung“ und „Fehlerhafte Ausführung“ könnten auch dahingehend erweitert werden, dass über einen Trigger (=Auslöser) E-Mail-Nachrichten an die für die Fehlerkorrektur zuständigen Personen (bei fehlerhafter Ausführung) oder an die Personen, die die aktuellen Daten benötigen (bei korrekter Ausführung), gesendet werden.

Für unser Beispiel wäre es am einfachsten, wenn wir mit dem Prozessfluss des Datenflusses *flow2* als Ausgangspunkt weiter arbeiten würden. Damit könnten wir dann weitere Elemente hinzufügen. Manchmal würde diese Vorgehensweise ausreichen.

Allerdings ist dies nicht immer der beste Weg. Stattdessen werde ich eine andere Methode verwenden, die modularer und flexibler ist.

1. Zunächst erstelle ich einen neuen Fluss.

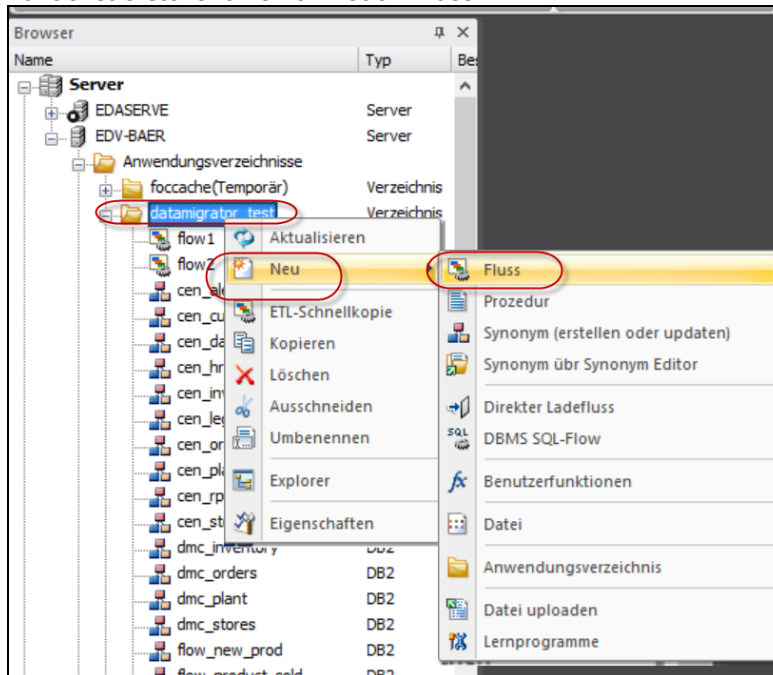


Abbildung 3: Einen neuen Fluss erstellen

2. Als nächstes wähle ich den Tab *Prozessfluss* und ziehe den bereits erstellten Fluß *flow2* auf den Arbeitsbereich.

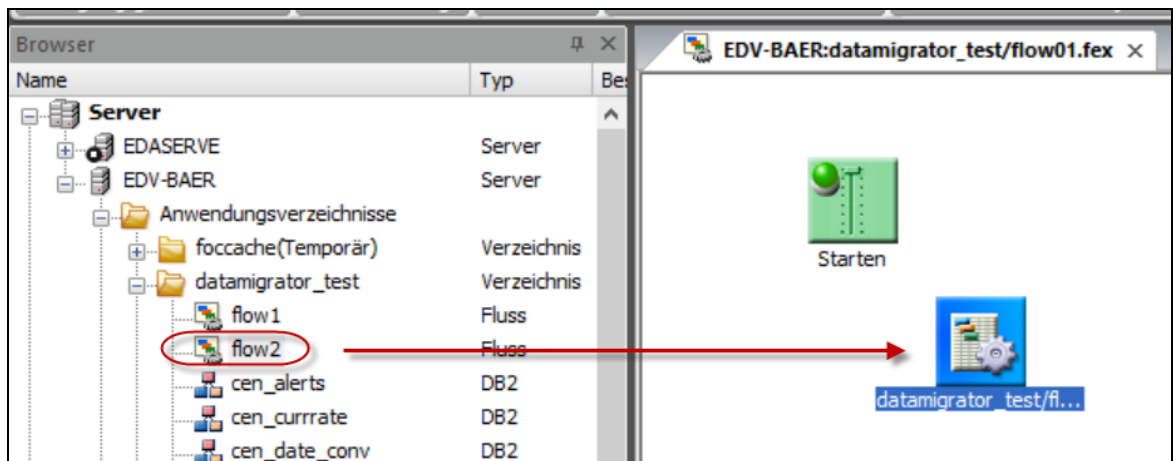


Abbildung 4: Fluss *flow2* hinzufügen

- Die Verbindung von *Starten* mit dem Fluss *flow2* erfolgt mit einem **rechten** Mausklick auf *Starten* und Ziehen auf *flow2*.

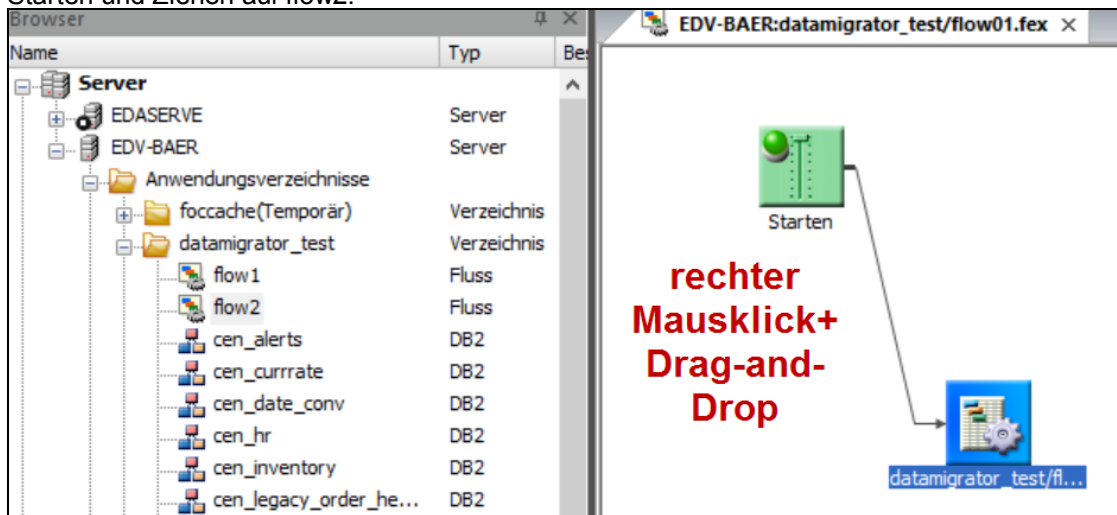


Abbildung 5: Verbinden der beiden Flüsse

Hinweis:

Bei Verwendung dieser Methode wird jeder Fluss separat gesichert. Ein Datenfluss-Objekt kann dadurch in mehr als einem Prozessfluss verwendet werden. Der Prozessfluss kann wie gewünscht angepasst werden, und verschiedene Datenfluss-Objekte können eingegliedert werden.

Vergessen Sie nicht, den Prozessfluss zu sichern. Ich habe dazu den Namen *flow2_process* verwendet.

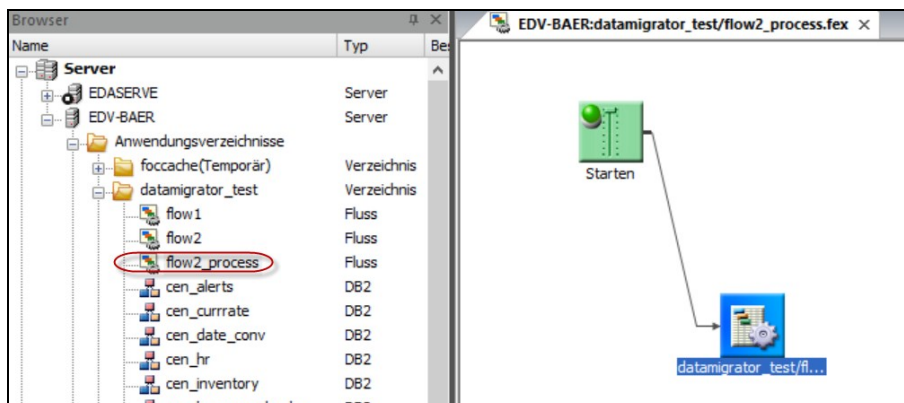


Abbildung 6: Neuer Prozessfluss *flow2_process*

Prozeduren und Bedingungen hinzufügen

Als nächstes erstelle ich zwei Prozeduren. Diese werden mit dem Datenfluss derartig verbunden, dass die eine ausgeführt wird, wenn der Datenfluss erfolgreich ausgeführt wurde. Die andere wird dann ausgeführt, wenn der Datenfluss nicht korrekt beendet wurde.

Gemäß dem Motto „Modular ist besser“ erstelle ich zwei Prozeduren als eigenständige Objekte, die dann sowohl im gerade zu erstellenden Fluss, als auch für andere verwendet werden können.

Über das Kontextmenü des Ordners *datamigrator_test* wähle ich jeweils Neu → Prozedur. Die erste Prozedur benenne ich *success*, die zweite *failure*. Der Inhalt der beiden Prozeduren ist in der folgenden Abbildung zu sehen.

-TYPE SUCCESS! (für Prozedur success)

-TYPE FAILURE! (für Prozedur failure).

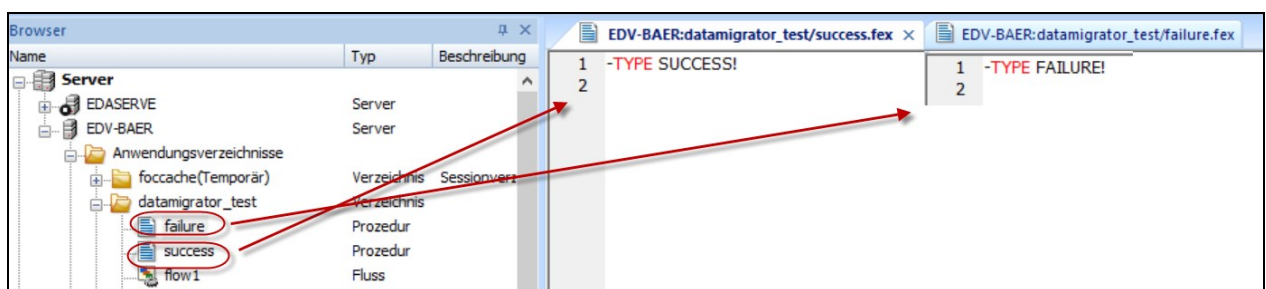


Abbildung 7: Die beiden Prozeduren *success* und *failure*

Nun ziehe ich die beiden Prozeduren aus dem Navigationsbaum auf den Arbeitsbereich rechts neben den Datenfluss *flow2*. Die Verbindung von *flow2* zu den beiden Prozeduren erfolgt wieder über einen rechten Mausklick auf *flow2* und durch Ziehen auf die Prozedur *success* beziehungsweise *failure*.

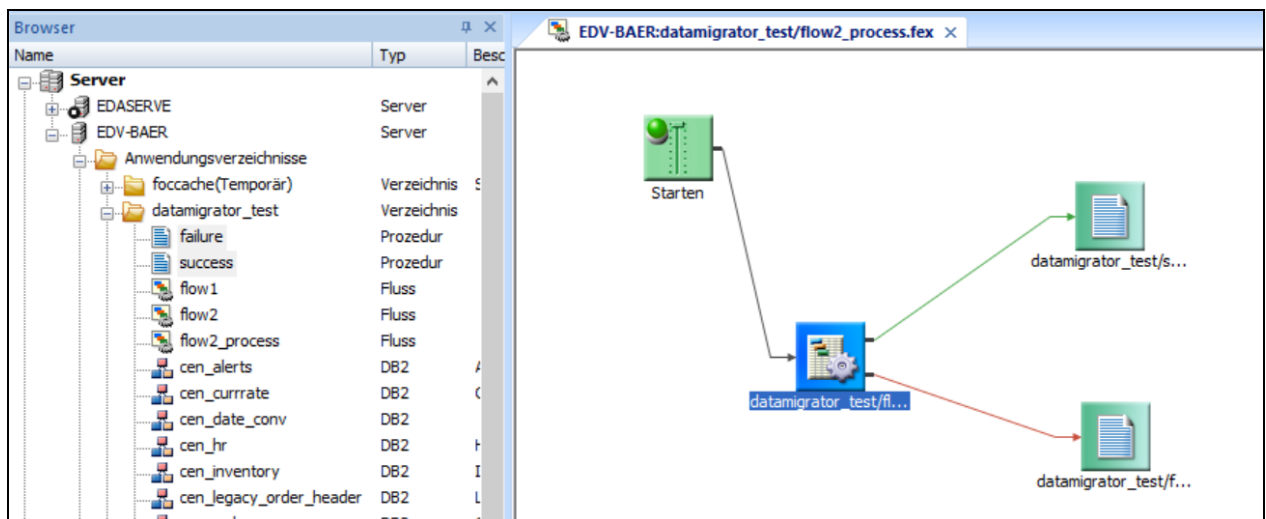


Abbildung 8: Prozessfluss mit den beiden Prozeduren *success* und *failure*

Hinweis:

Sicher haben Sie schon bemerkt, dass die Bezeichnung der einzelnen Objekte den Ordner-Namen *datamigrator_test* enthalten. Über das Kontextmenü eines Objektes und Auswahl Eigenschaften können Sie den Anzeigename ändern (siehe folgende Abbildung).

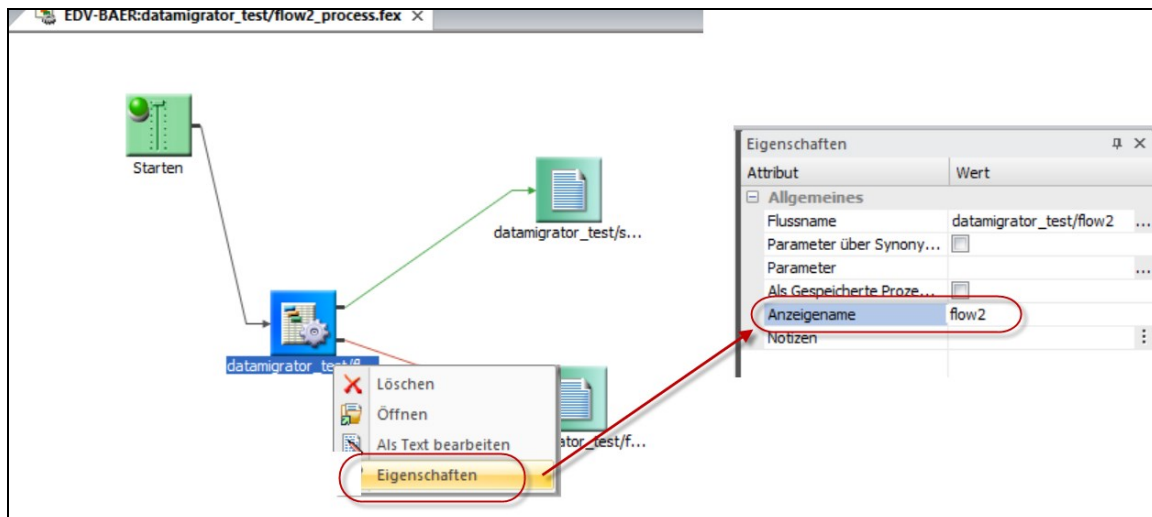


Abbildung 9: Ändern des Anzeigenamens über das Kontextmenü Eigenschaften

Die folgende Abbildung zeigt den Prozessfluss mit den geänderten Anzeigenamen.

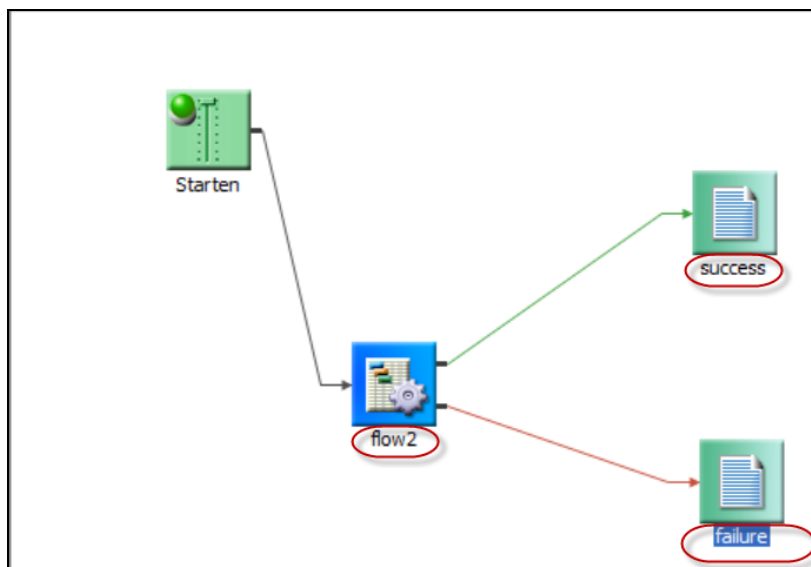


Abbildung 10: Prozessfluss mit geänderten Anzeigenamen

Wir haben zwar die beiden Prozeduren *success* und *failure* dem Fluss hinzugefügt, aber noch definiert, wie diese beiden Prozeduren reagieren sollen.

Durch einen Doppelklick auf den grünen bzw. roten Pfeil wird das Bedingungs Fenster geöffnet. Die bereits voreingestellten Default-Bedingungen können in diesem Fall übernommen werden.

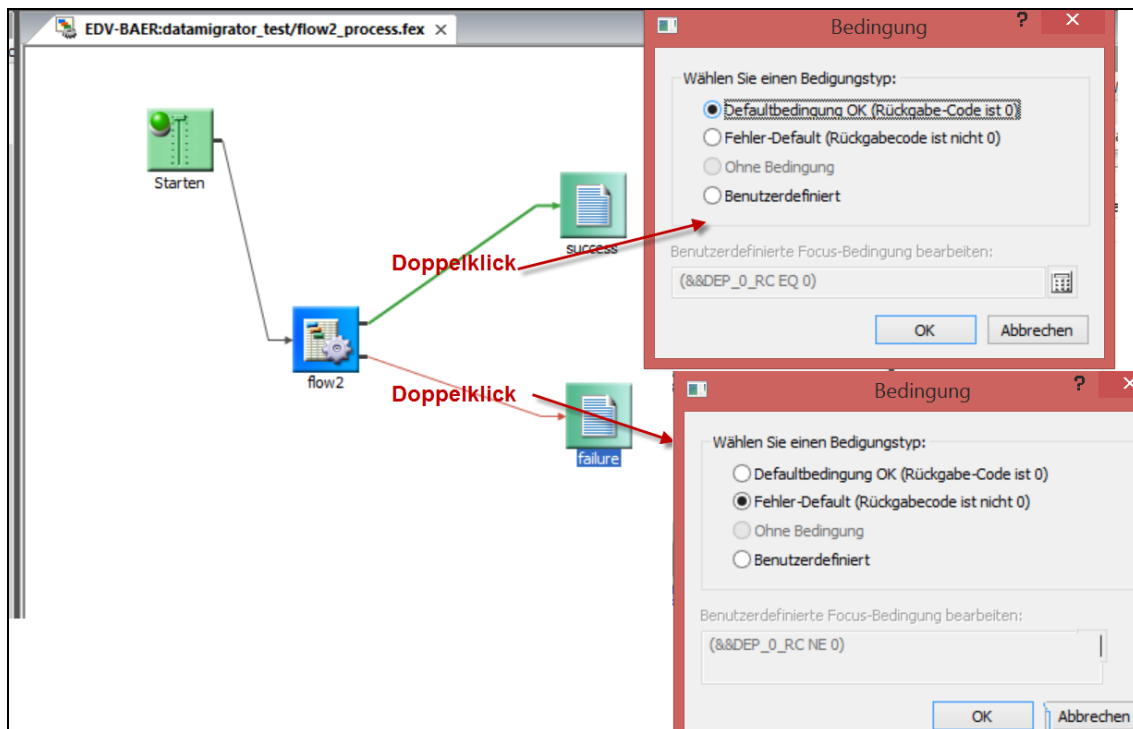


Abbildung 11: Einstellen der Bedingungen für die Prozeduren

Vergessen Sie nicht, Ihre Arbeit zu speichern.

Den Prozessfluss ausführen

Nun sollten wir den Fluss tatsächlich auch ausführen und prüfen, ob alles korrekt eingerichtet ist und die korrekten Meldungen angezeigt werden.

Dazu klicke ich auf das Icon *Ausführen*.

Sobald der Lauf beendet ist, *Letztes Log* anzeigen.

	Datetime	Message Code	Log Messages	Application	Name	Job
1	2015/02/19 00:36:30	(ICM18974)	----- Start of Log Record for datamigrator_test/flow2_process -----	datamigrator_test	flow2_process	20
2	2015/02/19 00:36:30	(ICM18122)	Request - datamigrator_test/flow2_process (Owner: bestgen) submitted.	datamigrator_test	flow2_process	20
3	2015/02/19 00:36:30	(ICM18027)	DEP_0: flow flow2 started.	datamigrator_test	flow2_process	20
4	2015/02/19 00:36:30	(ICM18016)	Request datamigrator_test/flow2 submitted. Please, wait for request to complete	datamigrator_test	flow2_process	20
5	2015/02/19 00:36:30	*	.	datamigrator_test	flow2_process	20
6	2015/02/19 00:36:30	(ICM18762)	Job ID: 20150219003630_297bf7f	datamigrator_test	flow2_process	20
7	2015/02/19 00:37:18	(ICM18763)	Request datamigrator_test/flow2 complete	datamigrator_test	flow2_process	20
8	2015/02/19 00:37:18	(ICM18039)	DEP_0 flow2 Return Code = 0	datamigrator_test	flow2_process	20
9	2015/02/19 00:37:18	(ICM18015)	DEP_1: procedure success started.	datamigrator_test	flow2_process	20
10	2015/02/19 00:37:18		SUCCESS!	datamigrator_test	flow2_process	20
11	2015/02/19 00:37:18	(ICM18039)	DEP_1 success Return Code = 0	datamigrator_test	flow2_process	20
12	2015/02/19 00:37:18	(ICM18031)	Finished	datamigrator_test	flow2_process	20
13	2015/02/19 00:37:18	(ICM18072)	Elapsed run time 0:00:48	datamigrator_test	flow2_process	20
14	2015/02/19 00:37:19	(ICM18975)	----- End of Log Record for datamigrator_test/flow2_process -----	datamigrator_test	flow2_process	20

Abbildung 12: Log mit Erfolgsmeldung SUCCESS

Wie erwartet, zeigt das Log an, dass der Ablauf korrekt durchgeführt wurde und damit die Prozedur *success* mit der Nachricht **SUCCESS!** ausgegeben wurde.

Vorschau:

Im nächsten Artikel 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