

Prozessabläufe in JDito und Frames

AID 070 DE für ADITO

ab Version 4.2

Gültig für den klassischen Client (Swing)



© 2014 ADITO Software GmbH

Diese Unterlagen wurden mit größtmöglicher Sorgfalt hergestellt. Dennoch kann für Fehler in den Beschreibungen und Erklärungen keine Haftung übernommen werden. Wir sind für Feedback zu den Themen, Inhalten, aber auch noch vorhandenen Fehlern dankbar und würden uns freuen, Ihre Meinung zu hören. Die in diesen Unterlagen enthaltenen Daten und Angaben, einschließlich URLs und anderer Verweise können ohne vorherige Ankündigung geändert werden. Alle in diesen Unterlagen aufgeführten Produkt- und Firmennamen sind unter Umständen Marken oder geschützte Zeichen der einzelnen Firmen. Ohne ausdrückliche schriftliche Einverständniserklärung der ADITO Software GmbH darf kein Teil dieses Dokumentes vervielfältigt oder in einer Datenverarbeitungsanlage gespeichert oder in diese eingelesen werden. Diese Einschränkung gilt unabhängig von Art und Weise der Datenerfassung.

Autor: MW, FA, KN. Version 10.3. Zuletzt geändert 05.09.2017

Version	Änderungen
10.3	Anpassung der Formatierungen
10.2	Screenshots erneuert
10.1	Hinweis zu Tabellenkomponenten und Image-Variablen
10.0	Anpassung an ADITO4.6
2.0	Letzter Stand vor Übernahme in Versionierung

Inhaltsverzeichnis

1.	Über dieses Dokument	4
1.1.	Alle ADITO-Versionen	4
1.2.	ADITO 4.2	4
1.2.1.	Berechnungsmodi	4
2.	Ablauflogik	5
2.1.	Hierarchie	5
2.2.	Komponenten	5
2.2.1.	Komponentenwertänderungen in ADITO bis ADITO4.1 und im Modus "Classic"	6
2.2.2.	Komponentenwertänderungen in ADITO ab ADITO 4.2	6
2.2.3.	Mögliche Auslöser für Wertänderungen	7
2.3.	In der Praxis	8
2.3.1.	Abhängigkeit von globalen Variablen	8
2.3.2.	Abhängig von einem Zeitstempel	9
2.3.3.	Abhängigkeit von einer Image-Variable.....	11
2.3.4.	Abhängigkeit von Tabellenwerten	11
2.3.5.	Kombinierte Abhängigkeiten	12
2.3.6.	Endlosschleifen (Strict)	13
2.3.7.	Endlosschleifen (Classic)	14
2.3.8.	Endlosschleifen (Compatible)	14
2.3.9.	Image-Variablen, die in Komponentenprozessen generiert werden.....	14

1. Über dieses Dokument

Frames (Bildschirmmasken) sind in ADITO4 online mehr als nur Oberflächenfelder für Datenbankspalten. Da sie sich auf veränderte Gegebenheiten im Frame anpassen müssen, können Komponenten in vielen Einzelprozessen auf Zustände des Frames oder andere Komponenten reagieren, sich also **neu berechnen**.

1.1. Alle ADITO-Versionen

Diese Neuberechnung hängt vom verwendeten Berechnungsmodus ab. Bei ADITO bis Version 4.1 wird nach einer Aktualisierung immer eine Neuberechnung vom **Client** angestoßen, wenn sich Komponenten als **nicht mehr gültig** („dirty“) ausweisen. Eine Neuberechnung kann auch vom Benutzer durch Klick auf **Aktualisieren** ausgelöst werden.

Da Komponenten nicht nur von einer, sondern unter Umständen von vielen Einflussgrößen abhängig sein können, ist eine Ablauflogik notwendig.

Dieses Dokument nennt die möglichen Prozesse und beschreibt diese Ablauflogik.

1.2. ADITO 4.2

Ab ADITO 4.2 wird beim Öffnen der Frames ein Ablaufgraph berechnet. Anhand dieses Ablaufbaums werden die Prozesse der Komponenten nacheinander ausgeführt. Diese Logik ist schneller als in vorhergehenden ADITO-Versionen (dies kann aus Kompatibilitätsgründen aber auch auf die klassische Logik umgestellt werden).

1.2.1. Berechnungsmodi

Die folgenden Berechnungsmodi können in ADITO4.2 als Server-Startparameter gesetzt werden:

- `Dadito.jdito.calculation.mode=STRICT` (Standard): Komponentenabhängigkeit ermittelt sich automatisch – die neue Logik.
- `Dadito.jdito.calculation.mode=CLASSIC` (Wie in ADITO 4.1 und älter): Komponentenberechnung nach Abhängigkeiten.
- `Dadito.jdito.calculation.mode=COMPATIBLE`: Wie strict, nur ohne Hinweismeldung bei Rekursionen.

2. Ablauflogik

2.1. Hierarchie

Im ADITO4 Client werden die Frames angezeigt, welche mit Komponenten bestückt sind. Client, Frames und andere Komponenten können auf Prozesse Einfluss nehmen.



Der Client gibt zur Laufzeit System- und globale Variablen aus. Der Frame behält die Kontrolle über die Image-Variablen und reagiert, sobald sich eine vom Client verwaltete Variable ändert. Die Komponente behält die Kontrolle über ihre eigenen Prozesse.

2.2. Komponenten

Unter Komponenten werden im Folgenden die Elemente eines Frames in ADITO4 online genannt. Der Frame, obwohl technisch gesehen selbst eine Komponente, wird als übergeordnetes Element angesehen.

Der Wert einer Komponente ist immer in der **Formel** hinterlegt. Nur die Änderung des Wertes kann überhaupt eine Neuberechnung anderer Komponenten anstoßen.

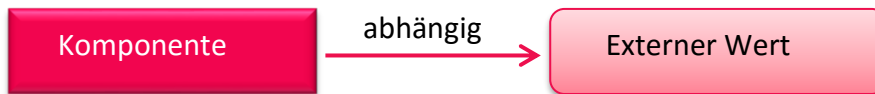
Folgende Komponenten besitzen die für die Neuberechnung notwendige Formel:

- Edit
- Label
- Memo
- Combobox
- Radiobutton
- Checkbox
- Liste
- Lookup
- Farbkomponente
- Visualisierung (für Tachometer / Thermometer Visualisierungen)

Ausnahmen zu dieser Regel werden weiter unten in Beispielen genannt.

2.2.1. Komponentenwertänderungen in ADITO bis ADITO4.1 und im Modus "Classic"

Eine Komponente wird dann neu berechnet, wenn sich ein Wert geändert hat, von dem die Komponente abhängig ist.



Ändert sich nun der externe Wert, wird die Komponentenneuberechnung angestoßen:



„dirty“ bedeutet, dass der Wert der Komponente nicht mehr gültig ist.



Abhängigkeiten greifen nur direkt in den Formel-Prozessen der Komponente, nicht in importierten Prozessen.

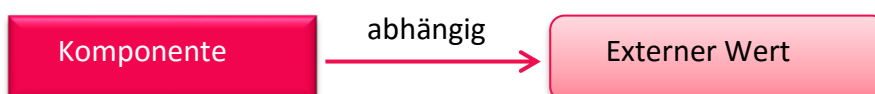


Eine Ausnahme ist der Baum. Dieser behält seinen Wert nicht, sondern setzt sich sofort dirty. Der Baum berechnet sich also bei jedem Update neu, auch wenn sich der Wert nicht geändert hat.

2.2.2. Komponentenwertänderungen in ADITO ab ADITO 4.2

Der grundlegende Ablauf der Komponentenwertänderung ist in ADITO 4.2 wie in den Vorgängerversionen.

Eine Komponente wird auch hier dann neu berechnet, wenn sich ein Wert geändert hat, von dem die Komponente abhängig ist.



Ändert sich nun der externe Wert, wird die Komponentenneuberechnung angestoßen:



„dirty“ bedeutet, dass der Wert der Komponente nicht mehr gültig ist.

2.2.2.1. Ablaufgraph

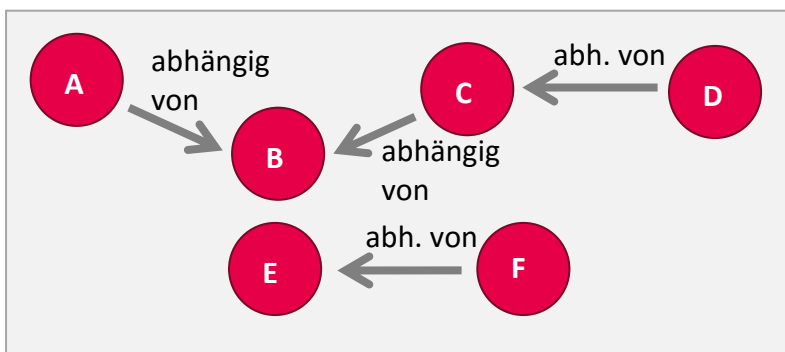
Alle Komponenten werden auf ihre Abhängigkeiten geprüft, sie berechnen sich dann, an welcher Stelle sie sich befinden. Sind Komponenten von keinen anderen Komponenten mehr abhängig, werden sie als erstes berechnet. Als nächstes folgen die Komponenten, die nur eine Abhängigkeit haben.

In diesem Fall wäre die Berechnungsabfolge:

1. **B**
2. **C und E**
3. **A, D und F**

Bei einer Wertänderung von B berechnen sich selbstverständlich nur A, C und D neu.

2.2.3. Mögliche Auslöser für Wertänderungen



Bei folgenden Änderungen wird ein Auslöser benötigt, bevor nach dem „dirty“-Setzen einer Variablen eine Neuberechnung stattfindet:

- Variablen („\$global.“, „\$image.“)
- Datumsvariablen („\$sys.date“, „\$sys.staticdate“)

Eine Eingabe eines Wertes in eine Komponente bzw. Setzen eines Wertes über `swing.setValue` oder Weiterblättern löst eine Neuberechnung aus, sofern sich der Wert geändert hat.



Erfolgt ein Klick auf Aktualisieren, so wird immer eine Neuberechnung angestoßen, auch wenn sich der Wert selbst nicht geändert hat.

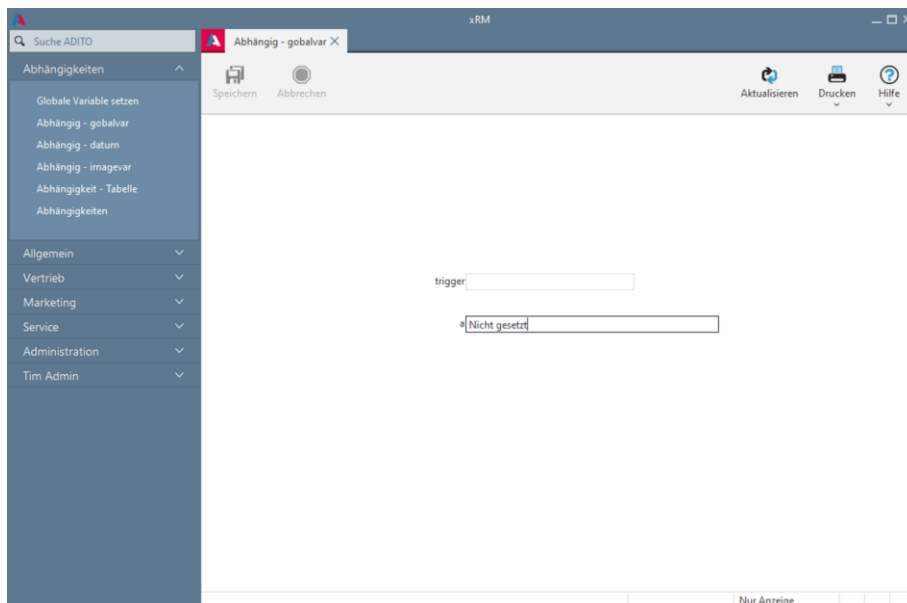
2.3. In der Praxis

Die Clients zeigen mitunter Clients von ADITO 4.1, die Beschreibungen sind aber auch für ADITO 4.2 gültig.

2.3.1. Abhängigkeit von globalen Variablen

Greift eine Komponente innerhalb eines Frames auf eine globale Variable zurück, so wird der Wert der Komponente sofort neu berechnet, wenn sie über einen Auslöser (Trigger) aktiviert wurde. Die Änderung an der globalen Variablen hat aber keinen Einfluss auf die Anzeige der Komponente.

1. Öffnen des Frames:



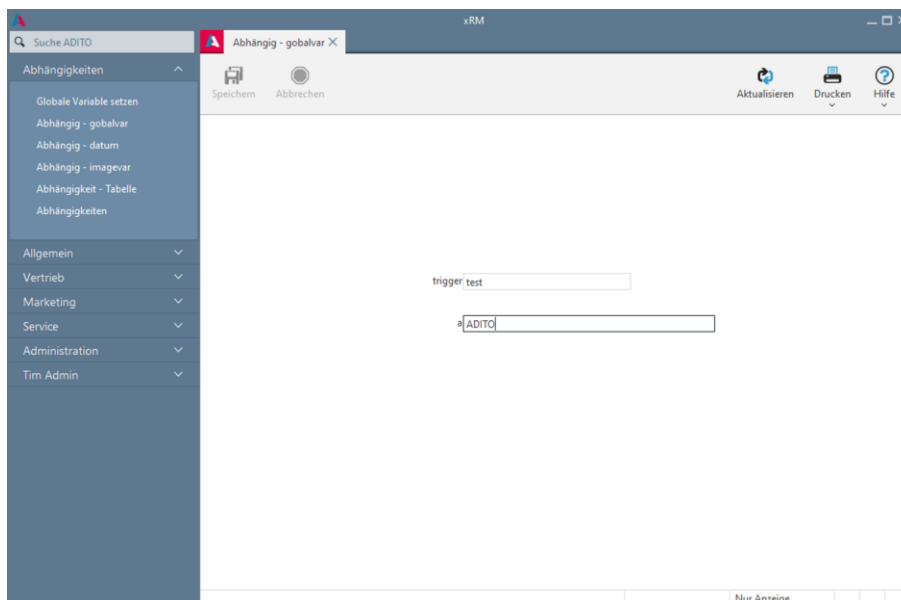
2. Ausgabe im Log:

```
-----
Frame Abhaengig - globalvar gestartet
Neuberechnung a
```


3. Setzen der globalen Variable:



Es gibt keine Ausgabe der Wertänderung im Log und keine korrigierte Anzeige. Würde man nun auf **Aktualisieren** klicken, so würde sich die Komponente sowieso neu berechnen. Da sich aber der Wert der globalen Variable geändert hat und die Komponente somit dirty gesetzt ist, bedarf es nur eines Auslösers, um die Neuberechnung in Gang zu setzen. Dieser Trigger ist eine völlig alleinstehende Edit-Komponente. Trägt man in diese etwas ein, werden Anzeige und Log-Ausgabe angestoßen.



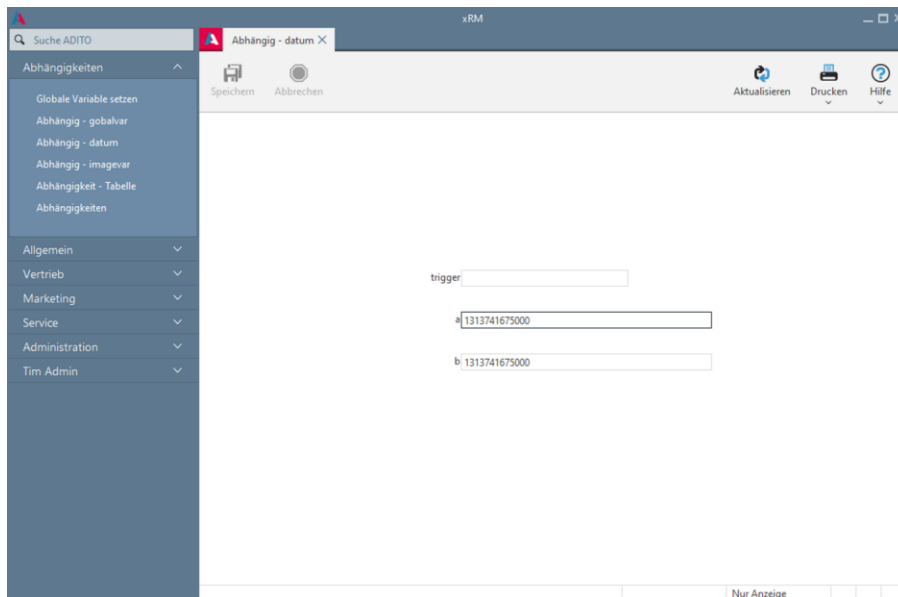
Neuberechnung a

2.3.2. Abhängig von einem Zeitstempel

Zeitstempel in ADITO4 online sind Long-Werte und per se immer Millisekunden genau. Das bedeutet, dass sich eine Komponente, welche von einem Zeitstempel gefüllt wird, bei jeder Berechnung einen neuen Wert registrieren und sich neu berechnen würde.

Die Systemvariable **\$sys.date**, obwohl von der Länge her Millisekunden genau, wird aber nur jede Sekunde aktualisiert. Dieser Variable gegenüber steht **\$sys.staticdate**, diese wird nur durch Benutzerkommando aktualisiert.

Ein Frame mit einer Komponente, welche auf beide Zeitstempel-Variablen zugreift, führt beim Start folgende Berechnungen aus:



```
-----
Frame Abhaengig - $sys.date gestartet
Neuberechnung b
Neuberechnung a
Neuberechnung a
Neuberechnung a
Neuberechnung a
Neuberechnung a
Neuberechnung a
```

Die Komponente **b** greift hierbei auf **\$sys.staticdate** zu, **a** auf **\$sys.date**.



Die Berechnung von `$sys.date` wird mehrfach ausgeführt.

Aktiviert man über den Trigger die Neuberechnung des Frames, so wird nur **a** aktualisiert, nicht aber **b**, da `$sys.staticdate` nur bei einem expliziten Neuberechnungskommando (Datensatz wechseln, Benutzer klickt auf **Aktualisieren**) ausgeführt wird.

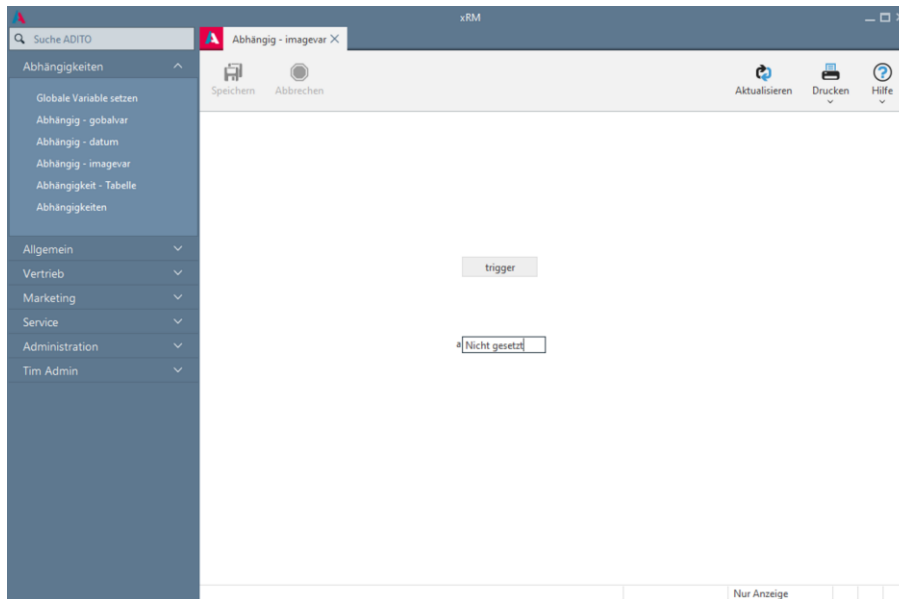
```
Neuberechnung a
Neuberechnung a
```

2.3.3. Abhängigkeit von einer Image-Variable

Im Gegensatz zu globalen Variablen bedingt eine Änderung der Image-Variable sofort die Neuberechnung und Anzeigeänderung der betroffenen Komponenten.

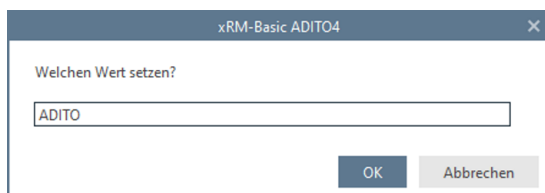
Wird ein Frame mit einer Komponente gestartet, welche direkt von einer Image-Variablen abhängig ist, wird erst einmal die Komponente berechnet:

Die Anzeige kann bei Änderung von Image-Variablen deshalb gleich aktualisiert werden, weil der Frame selbst über diese Variablen wacht.



 Frame Abhaengig - imagevar gestartet
 Neuberechnung a

Wird nun die Image-Variable geändert, so wird die Neuberechnung sofort in Gang gesetzt.

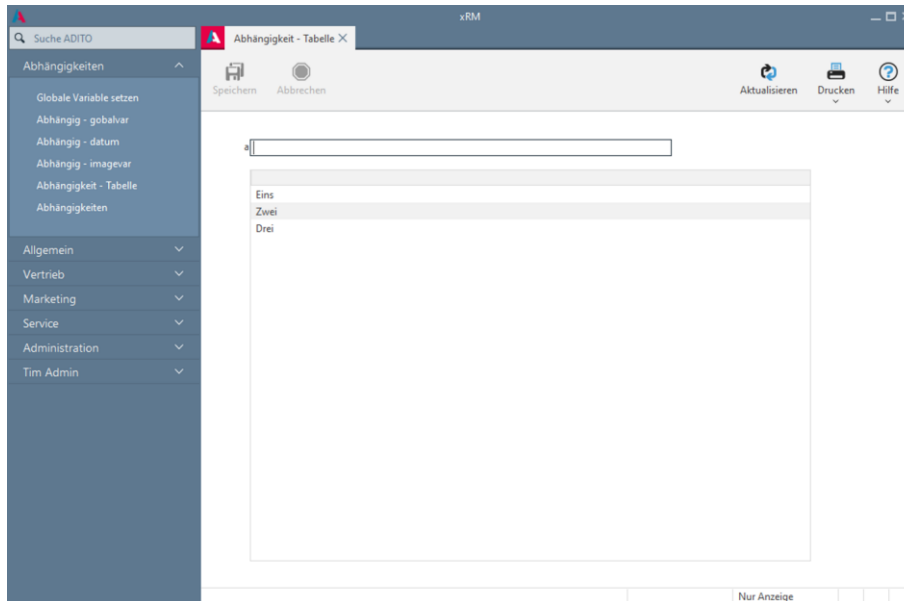


Neuberechnung a

2.3.4. Abhängigkeit von Tabellenwerten

Ist eine Komponente vom Wert einer Tabelle abhängig, so wird wie bei einer Image-Variablen automatisch eine Neuberechnung und neue Anzeige angestoßen.

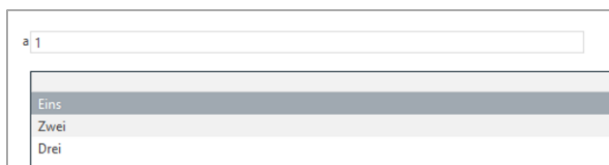
Beachten Sie bitte den Hinweis in [Kapitel 2.3.9!](#)



Ausgabe im Log:

```
-----
Abhaengigkeit - Tabelle
Neuberechnung a
```

Sobald sich die Markierung in der Tabelle ändert, wird die Anzeige neu berechnet.



```
Neuberechnung a
```



Die Einstellung „Automatisch“ bei der Wertermittlung kann bei unklarem Rückgabewert dazu führen, dass der Tabelle Füllen-Prozess mehrfach ausgeführt wird.

2.3.5. Kombinierte Abhängigkeiten

In der Praxis gibt es oft viele Komponenten auf einem Frame, die teilweise voneinander abhängig sind, ganz im Unterschied zu den vorhergegangenen, vereinfachten Beispielen.

2.3.5.1. In der Theorie (ADITO 4.2)

Gegeben sei ein Frame mit den Komponenten aus [Kapitel 2.2.2.1](#):

- **a**: Formel: b
- **b**:
- **c**: Formel: b
- **d**: Formel: c
- **e**:
- **f**: Formel: e

Beim Start des Frames zeigt sich folgende Ausgabe:

```
R-37-R-0000-S Projekt-Ausgabe. [ID 1] [->] ***** Vor Start
Abhaengigkeiten
R-37-R-0000-S Projekt-Ausgabe. [ID 1] [->] *** Neuberechnung b
R-37-R-0000-S Projekt-Ausgabe. [ID 1] [->] *** Neuberechnung e
R-37-R-0000-S Projekt-Ausgabe. [ID 1] [->] *** Neuberechnung c
R-37-R-0000-S Projekt-Ausgabe. [ID 1] [->] *** Neuberechnung d
R-37-R-0000-S Projekt-Ausgabe. [ID 1] [->] *** Neuberechnung f
R-37-R-0000-S Projekt-Ausgabe. [ID 1] [->] *** Neuberechnung a
R-37-R-0000-S Projekt-Ausgabe. [ID 1] [->] ***** Nach Start
Abhaengigkeiten
```

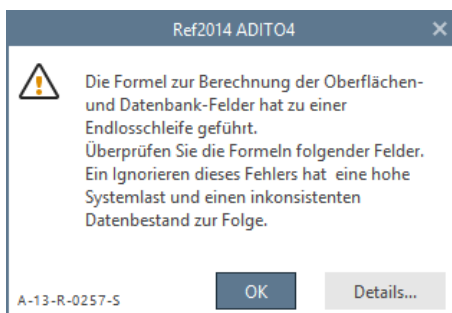
Wird nun in **b** ein veränderter Wert eingetragen, tritt folgender Ablauf in Kraft:

```
R-37-R-0000-S Projekt-Ausgabe. [ID 1] [->] *** Neuberechnung c
R-37-R-0000-S Projekt-Ausgabe. [ID 1] [->] *** Neuberechnung d
R-37-R-0000-S Projekt-Ausgabe. [ID 1] [->] *** Neuberechnung a
```

Der Wert von **c** wird vor **d** berechnet, damit D nicht mehrfach berechnet werden muss.

2.3.6. Endlosschleifen (Strict)

Liefert die Formel tatsächlich eine Endlosschleife (**b** ist abhängig von **a**, und umgekehrt), dann wird beim Aufbau des Graphen diese Schleife ermittelt und der Benutzer erhält sofort eine entsprechende Meldung:



Nach Auftreten einer solchen Meldung ist der Frame inkonsistent, Berechnungen werden nicht zu Ende geführt.

2.3.7. Endlosschleifen (Classic)

Liefert die Formel einer Komponente tatsächlich bei jeder Neuberechnung einen anderen Wert zurück, so wird die Ausführung und Berechnung nach zehn Durchläufen unterbrochen und es erscheint eine Fehlermeldung.

Ist beispielsweise in einer Formel folgender Code hinterlegt:

```
var ts = vars.getString("$sys.date");  
var uid = util.getNewUUID();  
result.string(ts + uid);
```

Dann führt diese Komponente beim Öffnen des Frames zum Fehler:

```
-----  
Abhaengigkeit - Endlosschleife  
Neuberechnung a  
Neuberechnung a  
Neuberechnung a  
Neuberechnung a  
Neuberechnung a  
Neuberechnung a  
Neuberechnung a  
Neuberechnung a  
Neuberechnung a  
Neuberechnung a  
Neuberechnung a  
A-13-R-0024-S Die Formel zur Berechnung der Oberflächen- und  
Datenbank-Felder hat zu einer Endlosschleife geführt. Überprüfen Sie  
die Formeln folgender Felder. Ein [ID 1] [->] comp.a
```



Wenn Sie Endlosschleifen von vorne hinein verhindern möchten, setzen Sie die Abhängigkeiten in die importierten Prozesse. Dann wird keine Mehrfachberechnung durchgeführt.

2.3.8. Endlosschleifen (Compatible)

Die Berechnung bei Compatible verhält sich wie bei "Strict" ausgeführt. Sobald eine Berechnung mehrfach ausgeführt wird, schaltet das System im Fall einer Schleife aber auf "Classic" und versucht, die Berechnung innerhalb von zehn Schritten durchzuführen.

2.3.9. Image-Variablen, die in Komponentenprozessen generiert werden.

Ist eine Tabelle abhängig von einer Image-Variablen, so wird diese nicht berechnet, wenn die Image-Variable wiederum im `valueProcess` einer Komponente berechnet wird. Die Neuberechnung wird hier nur durch einen Verweis auf einen Komponentenprozess angestoßen.