

KOMPAKT  
INFO

**ABAP**-Generierung  
aus Entscheidungstabellen  
mit LF-ET

## 1 Einleitung

Dieses Dokument bietet einen ersten kompakten Überblick, wie aus einer mit LF-ET erstellten Entscheidungstabelle ABAP-Programmcode generiert werden kann.

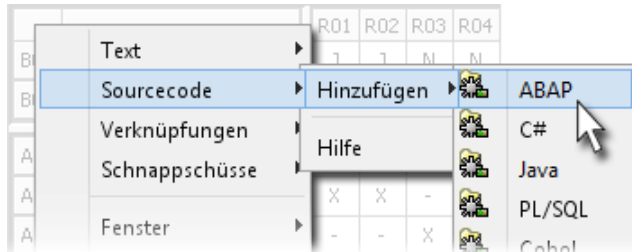
Die für das Beispiel gewählte Entscheidungstabelle ist ganz bewusst sehr einfach, damit man sich ganz auf das Thema „Code-Generierung“ konzentrieren kann:

Vergleich zweier Strings und Ergebnis-Rückgabe als Text				
		R01	R02	R03
B01	String 1 kleiner String 2 ?	J	N	N
B02	String 1 größer String 2 ?	-	J	N
A01	Ergebnis: String 1 ist kleiner als String 2	X	-	-
A02	Ergebnis: String 1 ist größer als String 2	-	X	-
A03	Ergebnis: String 1 ist gleich String 2	-	-	X

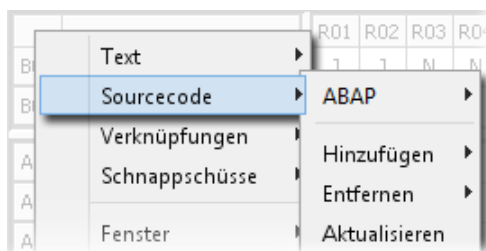
## 2 ABAP in der Entscheidungstabelle aktivieren

Bei jeder Entscheidungstabelle kann individuell festgelegt werden, für welche Programmiersprachen Programmcode generiert werden soll.

Mit der Funktion **Sourcecode > Hinzufügen** wird in der aktuell geöffneten Entscheidungstabelle eine weitere Programmiersprache aktiviert:



Alle aktivierten Programmiersprachen (hier ABAP) erscheinen automatisch unter dem Menüpunkt **Sourcecode** und bieten einen direkten Zugriff auf die einzelnen Source-Elemente der Entscheidungstabelle:



### 3 Einmalige Programmierung der Bedingungen und Aktionen

Damit aus einer Entscheidungstabelle Programmcode generiert werden kann, müssen für ABAP - **nur ein einziges Mal** - die folgenden Source-Elemente definiert werden:

- für jede **Bedingung** ein **logischer Ausdruck** und
- für jede **Aktion** ein (oder mehrere) **Statement(s)**

In unserem Beispiel werden die folgenden ABAP-Source-Elemente eingegeben:

	Fachliche Beschreibung	ABAP-Source-Element
Bedingung B01	String 1 kleiner String 2 ?	<code>( id_str1 &lt; id_str2 )</code>
Bedingung B02	String 1 größer String 2 ?	<code>( id_str1 &gt; id_str2 )</code>
Aktion A01	String 1 ist kleiner als String 2	<code>rd_result = mc_s1_lt_s2.</code>
Aktion A02	String 1 ist größer als String 2	<code>rd_result = mc_s1_gt_s2.</code>
Aktion A03	String 1 ist gleich String 2	<code>rd_result = mc_s1_eq_s2.</code>

#### **Anmerkungen:**

- Die Source-Elemente können (müssen aber nicht) direkt in der Entscheidungstabelle gespeichert werden, als zusätzliche Information zu den jeweiligen Bedingungen und Aktionen.
- Für die Bearbeitung der einzelnen Source-Elemente kann entweder direkt der in LF-ET integrierte Source-Editor, oder aber auch eine beliebige Entwicklungsumgebung, wie z.B. **NetWeaver**, verwendet werden.
- Die Klammern bei den logischen Ausdrücken sind nur eine Stil-Empfehlung.

## 4 Der generierte Sourcecode

Sobald in der Entscheidungstabelle alle Sourcecode-Elemente für die Bedingungen und Aktionen definiert sind, kann der „endgültige“ Programmcode generiert werden.

Hier der generierte Sourcecode, der in NetWeaver als INCLUDE-Programm „YLFET\_COMPARE\_INCL“ bereitgestellt ist:

```
Include      YLFET_COMPARE_INCL      aktiv

* Generated by LF-ET 2.1.4 (130212a)
* From decision table
* "D:/Projekte/LF-ET-Demo/ABAP/Compare.lfet"
* 18.03.2013 14:22
*
* Condition C01: String 1 kleiner String 2 ?
IF ( id_str1 < id_str2 ).
*   Rule R01 ---->
*
*   Action A01: Ergebnis: String 1 ist kleiner als String 2
*   rd_result = mc_s1_lt_s2.
*
*   Rule R01 <----
ELSE.
*
* Condition C02: String 1 größer String 2 ?
IF ( id_str1 > id_str2 ).
*   Rule R02 ---->
*
*   Action A02: Ergebnis: String 1 ist größer als String 2
*   rd_result = mc_s1_gt_s2.
*
*   Rule R02 <----
ELSE.
*   Rule R03 ---->
*
*   Action A03: Ergebnis: String 1 ist gleich String 2
*   rd_result = mc_s1_eq_s2.
*
*   Rule R03 <----
ENDIF.
ENDIF.
*
* End of generated ABAP source code
```

### ***Nur ein einziges Mal...***

...wurde in der Entscheidungstabelle manuell programmiert:

- zwei logische Argumente für die beiden Bedingungen
- drei Statements für die drei Aktionen

Diese Source-Elemente können vom Generator beliebig oft wiederverwendet werden.

Die geschachtelte Kontrollstruktur (der „IF-ELSE-Regelbaum“) wird bei jeder Generierung aus den Regeln der Entscheidungstabelle komplett neu berechnet und generiert.

Dies unterstützt leistungsstark die Entwickler und garantiert maschinell langfristig die exakte Übereinstimmung von Vorgabe und Programmcode.

### ***Starke Entlastung bei Wartung und Pflege:***

- Bei Änderungen an den Regeln - ein ganz typischer Änderungsfall - muss einfach nur neu generiert werden!
- Eine erneute Programmierung ist später nur „punktuell“ notwendig, wenn sich die Bedeutung einzelner Bedingungen oder Aktionen ändert, oder wenn neue Bedingungen oder Aktionen zur Entscheidungstabelle hinzukommen.

## **5 Anmerkungen zur Generierung**

### ***CASE-Statements***

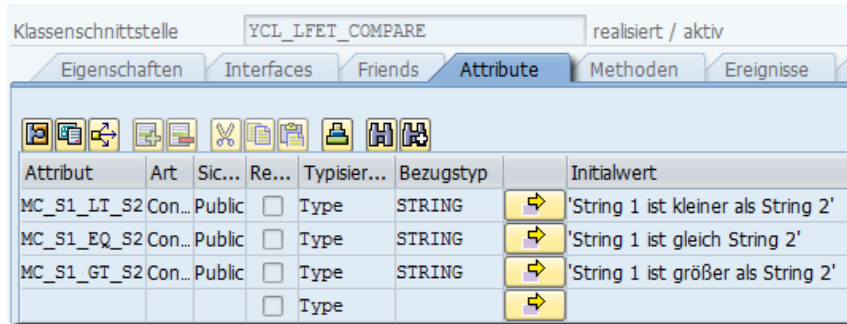
LF-ET bietet bei Bedarf auch die Möglichkeit, für einzelne Bedingungen anstelle eines IF-Statements ein CASE-Statement zu generieren.

### ***Änderungen am generierten Programmcode***

„Externe“ Änderungen am generierten Sourcecode, zum Beispiel verursacht durch Refactorings, kann LF-ET erkennen und die betroffenen Source-Elemente in der Entscheidungstabelle automatisch synchronisieren.

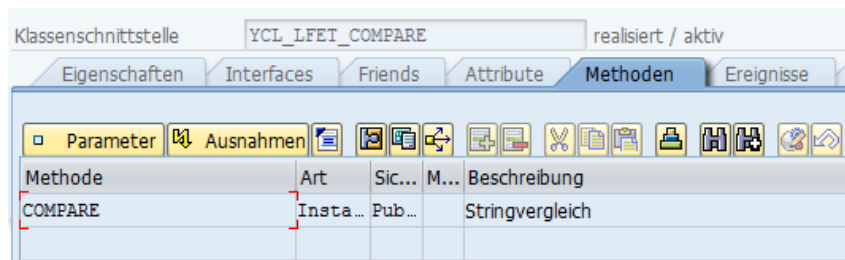
## 6 ABAP Integration

Der generierte Sourcecode wird als Methode „COMPARE“ in einer Klasse „YCL\_LFET\_COMPARE“ implementiert. Dazu wurde in ABAP die entsprechende Klasse definiert, mit den benötigten Konstanten...

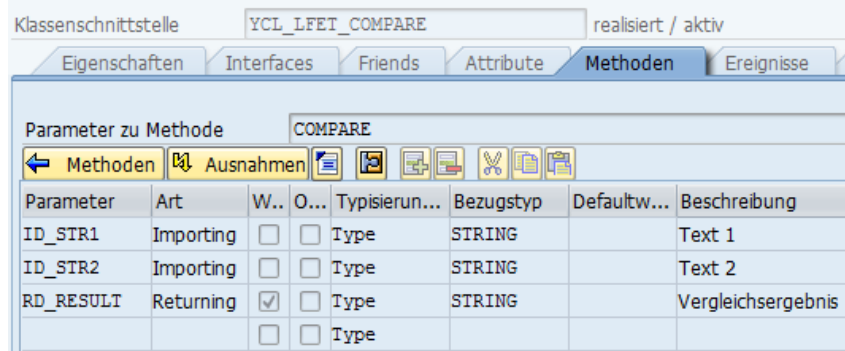


Attribut	Art	Sic...	Re...	Typisier...	Bezugstyp	Initialwert
MC_S1_LT_S2	Con...	Public	<input type="checkbox"/>	Type	STRING	'String 1 ist kleiner als String 2'
MC_S1_EQ_S2	Con...	Public	<input type="checkbox"/>	Type	STRING	'String 1 ist gleich String 2'
MC_S1_GT_S2	Con...	Public	<input type="checkbox"/>	Type	STRING	'String 1 ist größer als String 2'
			<input type="checkbox"/>	Type		

...und die Methode „COMPARE“ mit ihren Parametern:

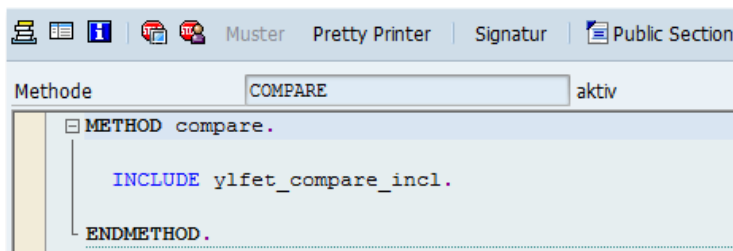


Methode	Art	Sic...	M...	Beschreibung
COMPARE	Insta...	Pub...		Stringvergleich



Parameter	Art	W..	O...	Typisierun...	Bezugstyp	Defaultw...	Beschreibung
ID_STR1	Importing	<input type="checkbox"/>	<input type="checkbox"/>	Type	STRING		Text 1
ID_STR2	Importing	<input type="checkbox"/>	<input type="checkbox"/>	Type	STRING		Text 2
RD_RESULT	Returning	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Type	STRING		Vergleichsergebnis
		<input type="checkbox"/>	<input type="checkbox"/>	Type			

Die Methode „COMPARE“ enthält nur ein INCLUDE-Statement, durch welches das generierte INCLUDE-Programm „YLFET\_COMPARE\_INCL“ in die Methode eingebunden wird:



```
Method: COMPARE (aktiv)
METHOD compare.
    INCLUDE ylfet_compare_incl.
ENDMETHOD.
```

Auf dieselbe Weise könnte das generierte INCLUDE-Programm problemlos auch in eine FUNCTION oder in ein FORM eingebunden werden.

## 6 Weiterführende Themen

Dieses Dokument erläutert nur ganz grob, im Sinne einer ersten Übersicht, die wichtigsten Grundprinzipien und Mechanismen der Programm-Generierung mit LF-ET für ABAP.

Weiterführende Themen wären beispielsweise:

- Komplexe Bedingungen, z.B. mit Unterprogrammaufrufen oder Datenbank-Zugriffen
- Umsetzung von Schleifen-Verarbeitungen, ggf. auch mehrfach geschachtelt
- Modellierung von Automaten
- Verschachtelung von Entscheidungstabellen, d.h. Entscheidungstabellen als Bedingungen oder Aktionen in anderen Entscheidungstabellen
- Automatische Protokollierung der Regel-Ausführungen
- Statistische Auswertung der Regel-Ausführungen, auch über längere Zeiträume
- Sichere Ableitung aller notwendigen Testfälle direkt aus den Entscheidungstabellen
- Automatische Ermittlung der Test-Deckungsgrade
- Unterstützung weiterer Sprachen, wie z.B.: C#, Java, PL/SQL, Natural, Cobol, VB, VBA, C++, C, Delphi, PHP5,...
- Anpassung der Generierung an spezielle Projekt-Bedürfnisse
- u.v.m.

## 7 Zusammenfassung

Entscheidungstabellen nach DIN 66241 helfen, fachliche Vorgaben leichter, präziser und vollständiger zu beschreiben. Der Umgang mit Entscheidungstabellen ist leicht erlernbar und LF-ET ermöglicht, mit seiner Excel-ähnlichen Oberfläche und zahlreichen leistungsstarken Funktionen, die leichte Beherrschung auch sehr komplexer Regelwerke.

Durch den Einsatz von Generatoren kann maschinell garantiert werden, dass die Software exakt mit den Vorgaben übereinstimmt. Auch der tatsächlich anfallende Programmieraufwand kann dadurch stark reduziert werden.

Insbesondere bei späterer Wartung und Pflege sind erfahrungsgemäß erhebliche Aufwands-Einsparungen zu erwarten, bei gleichzeitig deutlich reduziertem Fehler-Risiko.

Bei Bedarf kann LF-ET aus einer Entscheidungstabelle auch Programmcode für mehrere Programmiersprachen erzeugen.

LF-ET ist ein reines Entwicklungswerkzeug, zur Ausführung der generierten Programme ist **keinerlei zusätzliche Software erforderlich**.

## 8 Kontakt

Als IT-Beratungsunternehmen und -Dienstleister unterstützen wir seit vielen Jahren zahlreiche Behörden und Unternehmen aus den unterschiedlichsten Branchen.

Unabhängig vom technischen Umfeld ist dabei die Entscheidungstabelle ein wesentliches Hilfsmittel, um leichter und präziser Prozessabläufe beschreiben, implementieren und testen zu können.

Gerne stehen wir für weitere Informationen zur Verfügung.

**LOHRFINK**  
software engineering GmbH & Co. KG

Marie-Curie-Str. 6  
D-70736 Fellbach

Telefon 0711/3424 897-0  
Telefax 0711/3424 897-15  
info@lohrfink.de  
www.lohrfink.de  
www.lohrfink.de/lf-et