

# Fujitsu Software BS2000 ESQL-COBOL

## Version 3.0D

### Dezember 2013



## Readme-Datei

---

Alle Rechte vorbehalten, insbesondere gewerbliche Schutzrechte. Änderung von technischen Daten sowie Lieferbarkeit vorbehalten. Haftung oder Garantie für Vollständigkeit, Aktualität und Richtigkeit der angegebenen Daten und Abbildungen ausgeschlossen. Wiedergegebene Bezeichnungen können Marken und/oder Urheberrechte sein, deren Benutzung durch Dritte für eigene Zwecke die Rechte der Inhaber verletzen kann.

Copyright © 2025 Fujitsu

Die Marke Fujitsu und das Fujitsu Logo sind registrierte Marken der Fujitsu Limited, Japan in Europa und in anderen Ländern.  
BS2000 ist eine Marke der Fujitsu Germany GmbH in Europa.

<b>Einleitung</b>	<b>3</b>
<b>1 Free Reference Format</b>	<b>4</b>
1.1.1 Die Zeilenlänge	4
1.1.2 Generierte Kommentare	4
1.1.2.1 Auskommentierte SQL-Anweisungen	5
1.1.2.2 Fehlermeldungen	5
1.1.2.3 Info-Blöcke im Hosttext	5
1.1.3 Generierte Calls in COBOL	5
1.1.4 Dialekt-Abhängigkeit	6
1.1.5 SQLCA	6
<b>2 Ergänzung des Kap. 2.6 SQL-Kommentare in einem ESQL-COBOL-Programm</b>	<b>7</b>

Einleitung

=====

Beschrieben werden die Änderungen/Erweiterungen gegenüber ESQL-COBOL V3.0A und ESQL-COBOL für SESAM/SQL-Server.

Die Änderungen/Erweiterungen ergänzen das Handbuch

ESQL-COBOL V3.0A  
ESQL-COBOL für SESAM/SQL-Server  
Ausgabe November 2006  
Bestellnummer: U22424-J-Z125-3

Folgende Änderungen sind in dieser Readme-Datei beschrieben:

Zusammenstellung der Funktionalitätserweiterungen für ESQL-COBOL für SESAM/SQL-Server bezüglich des Free Reference Formats.

Erweiterung des Kapitels 2.6 „SQL-Kommentare in einem ESQL-COBOL-Programm“.

Schlüsselworte: COBOL2000, SQL, SESAM, Free Reference Format, C-style-Kommentar.

Diese und andere aktuelle Readme-Dateien sind online verfügbar unter <https://bs2manuals.ts.fujitsu.com/index>.

## 1 Free Reference Format

=====

COBOL 2000 ermöglicht das sogenannte Free Reference Format für COBOL-Text, das Beschränkungen der Zeilenlänge und die klassische Einteilung der Source-Text-Zeile in COBOL aufhebt. Der ESQL Precompiler kannte dieses Format in der Version V02.0 noch nicht. Diese Einschränkung wird in Version V03.0B aufgehoben, wobei die bisherige Funktionalität des Precompiler nicht verändert wird. Insbesondere verändert sich das Verhalten des Precompilers für Programme im klassischen Fixed Reference Format überhaupt nicht.

### 1.1 Umfang der Implementierung

=====

Die Funktionalität Free Format wird nicht durch eine externe Steuerung, sondern einzig und allein durch die sogenannten Compiler-Direktiven im Source-Text selbst ein- und ausgeschaltet. Insbesondere ist es nicht möglich, dass die Interpretation des COBOL-Textes im Free Format anfängt.

Dieses Verhalten ist identisch mit den Verhalten des COBOL2000-Compilers.

Die Funktionalität des Precompilers geht aber noch weiter: Auch der eingebettete SQL-Text unterliegt nicht mehr den Randbedingungen der früheren Precompiler Version, wo SQL-Text ebenfalls auf die Spaltenpositionen 8 bis 72 beschränkt war. SQL-Text kann im Free-Format nun auf denselben Spalten wie der COBOL-Text geschrieben werden. Einzelheiten und Einschränkungen werden weiter unten angegeben.

#### 1.1.1 Die Zeilenlänge

=====

Im Free-Format-Modus akzeptiert der Precompiler eine Zeilenlänge von 248. Das heißt, dass sowohl COBOL-Text als auch SQL-Text auf den Spaltenpositionen 1 bis 248 enthalten sein dürfen.

Der generierte Hosttext kann ebenfalls Zeilen bis zu höchstens dieser Länge enthalten.

Die im COBOL-Standard festgesetzte Zeilenlänge von 256 Spaltenpositionen bleibt bei beiden Produkten einer späteren Version vorbehalten.

#### 1.1.2 Generierte Kommentare

=====

Der Precompiler generiert in folgenden Situationen COBOL-Kommentare im erzeugten Hosttext:

- Auskommentierte SQL-Anweisungen
- Fehlermeldungen
- Info-Blöcke im Hosttext.

#### 1.1.2.1 Auskommentierte SQL-Anweisungen

=====

Die auskommentierten SQL-Anweisungen sehen im Fixed Format so aus wie bisher. Im Free-Format werden aus ihnen Free-Format-Kommentare erzeugt, indem das Kommentareinleitungssymbol ">" auf die ersten beiden Spaltenpositionen gesetzt wird.

Dabei werden gegebenenfalls Zeichen des SQL-Textes in Spaltenposition 1 und 2 überschrieben!

#### 1.1.2.2 Fehlermeldungen

=====

Fehlermeldungen bestehen aus zwei Bestandteilen: Ein Zeilenlineal, das die Spaltenposition eines Fehlers mit "...-----#-----..." markiert, und einem Fehlertext.

Der Fehlertext ist in beiden Formaten derselbe und als für beide Formate gültiger Kommentar ausgeführt. Dies wird erreicht, indem die Zeichenkette ">" genau auf Spaltenposition 7 und 8 positioniert wird.

Das Zeilenlineal für die Fehlerposition wird mit ">" in Spaltenposition 1 und 2 markiert. Wenn ein Fehler die "#" -Markierung in Position 1 oder 2 erzwingt, dann ist das "#" nicht zu sehen! (Aus diesem Grunde empfiehlt es sich, sich in Spaltenposition 1 und 2 beim Fehlermachen zurückzuhalten.) Die Fehlermeldungen tauchen aber in jedem Fall auf.

#### 1.1.2.3 Info-Blöcke im Hosttext

=====

Die vom Precompiler generierten Info-Blöcke sind wieder als für beide Formate gültiger Kommentar ausgeführt.

#### 1.1.3 Generierte Calls in COBOL

=====

Für SQL-Statements werden COBOL-Call-Anweisungen erzeugt. Die Parameterleisten, die für diese Call-Anweisungen erzeugt werden, nutzen im Free Format die volle Zeilenlänge aus. Wenn eine Zeile nicht ausreicht, erfolgt ein Zeilenumbruch, sooft es nötig ist.

Im Gegensatz zum Fixed Format, wo dieser Zeilenumbruch eventuell in der Mitte eines Wortes unter Erzeugung einer Fortsetzungsmarkierung in Spalte 7 erfolgen kann, wird im Free Format nur auf Whitespace umgebrochen. Der Grund liegt darin, dass die Trennung inmitten eines Wortes in zukünftigen Versionen des COBOL-Standards eventuell nicht mehr möglich ist.

Eine Methode, den Zeilenumbruch bei Parameterleisten auch im Fixed Format auf Whitespace zu erzwingen, wird nicht angeboten und ist auch nicht nötig, da jede ESQL-Anweisung in eine Free/Fixed Format Klammerung gesetzt werden kann.

#### 1.1.4 Dialekt-Abhängigkeit

=====

Es gibt keine Dialekt-Anhängigkeit. Die Free Format Funktionalität steht immer zur Verfügung, egal, welche der SDF-Optionen

```
... ,ESQL-DIALECT = *ALL-FEATURES(...)
```

oder

```
... ,ESQL-DIALECT = *ISO(...)
```

oder

```
... ,ESQL-DIALECT = *OLD(...)
```

für die Source-Properties angegeben wird.

#### 1.1.5 SQLCA

=====

Das SQLCA-Include-Element aus dem Lieferumfang kann gleichermaßen in beiden Reference-Formaten verwendet werden.

#### 1.3 Einschränkungen der Implementierung

=====

Ein "EXEC SQL" darf einem "END-EXEC" nicht auf derselben Zeile folgen.

Eine geringfügige Einschränkung ergibt sich daraus, dass es dem Precompiler völlig egal ist, ob er mit COBOL85 oder mit COBOL2000 zusammen verwendet wird. Nicht mit COBOL85 kompatiblen Sprachelemente (wie etwa das Free Format) werden nun nicht zurückgewiesen, so dass der COBOL85-Compiler Gelegenheit zum Flaggen eines Fehlers erhält.

Der Precompiler kann nicht erkennen, ob bei Free Format Source Text lediglich die Free Format Direktive vergessen wurde. Wegen der Behandlung der Spalten-Positionen 1 bis 7 und der Kürzung der Zeilen auf insgesamt 80 Spalten-Positionen kann der erzeugte Host-Text dem Originalprogramm sehr unähnlich sehen. In seltenen Fällen kann sogar legaler Source Text erzeugt werden, der aber nicht mehr dem intendierten Original entspricht!

#### 1.4 Glossar

=====

##### \* Free Reference Format

deutsch "Freies Referenz Format", oft kurz "Free Format": Die in dieser Version neu unterstützte Funktionalität. Gemeint ist nicht nur das durch den aktuellen COBOL-Standard vorgegebene freie Referenz-Format, sondern auch das analoge Zeilen-Layout von SQL.

##### \* Fixed Reference Format

deutsch "Festes Referenz Format", oft kurz "Fixed Format", auch als das "klassische COBOL Format" bezeichnet: das seit Jahrzehnten übliche Zeilenformat von COBOL.

## 2 Ergänzung des Kap. 2.6 SQL-Kommentare in einem ESQL-COBOL-Programm

=====

Seit ESQL-COBOL V3.0D können in SQL-Anweisungen auch C-style-Kommentare verwendet werden. Es sind geklammerte Kommentare, die mit /\* beginnen und mit \*/ enden und auch geschachtelt werden können.

Einschränkungen der Implementierung:

- C-style-Kommentare sind in den SQL-Anweisungen WHENEVER, INCLUDE und in DECLARE SECTION nicht erlaubt.
- END-EXEC kann nicht durch ein C-style-Kommentar innerhalb einer SQL-Anweisung auskommentiert werden. Es wird vom Precompiler immer als Ende der SQL-Anweisung interpretiert, auch wenn die eröffnenden Kommentarzeichen /\* vorausgehen.