

## Vorwort zur vierten Auflage

Dies ist die neueste Ausgabe der ABAP-Referenz und damit eine vollständige Beschreibung der Programmiersprache ABAP. Diese Auflage behandelt die Programmiersprache ABAP bis einschließlich Release 7.50. Das Buch ist eine gedruckte Form der ABAP-Schlüsselwortdokumentation, die als Ergänzung der verschiedenen Onlineformate dient. Die Buchform ermöglicht einen integralen Zugang zu dieser Dokumentation und soll dazu einladen, sich einen besseren Überblick über die ABAP-Anweisungen und ihre Zusammenhänge zu verschaffen, als es beim punktuellen Aufruf der Onlinedokumentation möglich ist.

Da im Vergleich zur Voraufgabe, die ABAP bis einschließlich Release 7.31 (gleichbedeutend zu Release 7.02) beschreibt, viele neue Themen dazugekommen sind, mussten an anderen Stellen Streichungen vorgenommen werden, damit der Inhalt noch zwischen zwei Buchdeckel passt. Diese Inhalte finden Sie bei Bedarf natürlich nach wie vor in der ABAP-Onlinedokumentation.

Mein Dank für diese Auflage gebührt vornehmlich Frau Janina Karrasch vom Rheinwerk Verlag, ohne die es das vorliegende Buch wahrscheinlich nicht gäbe. Sie hat das Projekt »Neuaufgabe der ABAP-Referenz« nicht nur angeschoben und betreut, sondern auch selbst die wesentliche Arbeit der Übertragung und Anpassung der Online-Inhalte in das Buchformat vorgenommen. Gemessen an der Komplexität der Sprache ABAP und damit auch des Buchinhalts wahrlich keine leichte Aufgabe, die Frau Karrasch auf höchstem Niveau gelöst hat. Zusätzlich musste sie den notorisch mit anderen Aufgaben beschäftigten eigentlichen Autor des Buches, also mich, dazu bringen, endlich auch seine Beiträge zu leisten. Sorry dafür! Aber jetzt ist es geschafft. Sie halten eine neue ABAP-Referenz in Händen, und wir hoffen, dass sie Ihnen von Nutzen ist.

**Horst Keller**

Knowledge Architect, PI Tech Core ABAP Server (SAP SE)

## Einleitung

Dieses Buch enthält eine vollständige Beschreibung der zur Verwendung freigegebenen Sprach-elemente der Programmiersprache ABAP, organisiert nach ihrem thematischen Zusammen-hang. ABAP wurde von SAP entwickelt und ist in SAP NetWeaver die Programmierschnitt-stelle des Application Server ABAP (AS ABAP). Als 4GL-Sprache ist ABAP speziell für die Massendatenverarbeitung in kommerziellen Anwendungen vorgesehen. Im Vergleich zu ele-mentaren Sprachen, in denen solche Funktionen in Bibliotheken liegen, bietet ABAP einen als *Open SQL* in die Sprache integrierten Datenbankzugriff sowie interne Tabellen für die dyna-mische Speicherung und die Bearbeitung tabellarischer Massendaten im Arbeitsspeicher. Außerdem verfügt ABAP über in die Sprache integrierte Schnittstellen zu anderen Program-mierumgebungen über Remote Function Call (RFC) oder zu anderen Formaten wie XML und JSON. Die Integration solcher Funktionen in die Sprache ist vor allem vorteilhaft für die stati-sche Überprüfbarkeit und die Performance von Programmen. Im Gegenzug enthält ABAP dadurch auch wesentlich mehr Sprachelemente als eine elementare Programmiersprache.

ABAP unterstützt ein auf Klassen und Interfaces basierendes objektorientiertes Programmier-modell. Aus Kompatibilitätsgründen wird auch noch eine auf Unterprogrammen und Funkti-onsbausteinen basierende prozedurale Programmierung unterstützt. Alle zugehörigen Sprach-elemente sind in der vorliegenden ABAP-Referenz aufgeführt. Diese Beschreibung umfasst auch die obsoleten Sprachelemente, deren Verwendung in neuen Programmen zwar nicht mehr empfohlen wird, die aber noch in vorhandenen Programmen vorkommen. Nicht darge-stellt sind dagegen die Sprachelemente, die nur zur internen Verwendung in Systemprogram-men vorgesehen sind. Die ABAP-Referenz dient somit als Nachschlagewerk für den täglichen Gebrauch beim Umgang mit der Sprache ABAP. Zur Verdeutlichung von Syntax und Semantik von ABAP-Anweisungen sind in der ABAP-Referenz auch Programme und Programmaus-schnitte dargestellt. Diese Syntaxbeispiele sind nicht zur direkten Verwendung in einer pro-duktiven Systemumgebung vorgesehen. Die Quelltexte der Beispiele dienen primär der bes-seren Erklärung und Visualisierung von Syntax und Semantik von ABAP-Anweisungen und nicht der Lösung konkreter Programmieraufgaben. Für produktive Anwendungsprogramme sollte daher immer eine einzelfallgerechte Lösung erarbeitet werden.

Hinweise zur Verwendung von ABAP wurden im E-Book *ABAP-Programmierichtlinien* (SAP PRESS 2009) zusammengestellt. Zu den Themen, für die eine solche Programmierichtlinie relevant ist, finden Sie diese in der vorliegenden ABAP-Referenz in knapper Form.

### Behandelte Releases

Die Basis der vorliegenden ABAP-Referenz ist der SAP NetWeaver Application Server, Release 7.50. Dies ist das derzeit neueste öffentlich verfügbare Release. Änderungen, die zwischen Release 7.02 (gleichbedeutend zu Release 7.31), d. h. dem Release, das in der letzten Auflage der ABAP-Referenz beschrieben wurde, und Release 7.50 erfolgten, sind in der vorliegenden Auflage gekennzeichnet:

- ▶ **7.40** Neuerungen, die es zu Release 7.40 gibt, sind am Seitenrand mit einem entsprechenden Icon gekennzeichnet. Dabei sind die mit den Support Packages SPO2, SPO5 und SPO8 ausgelieferten Neuerungen zusammengefasst. Für eine Auflistung der Neuerungen zu jedem einzelnen Support Package sei auf die Onlinedokumentation verwiesen.
- ▶ **7.50** Neuerungen, die es zu Release 7.50 gibt, sind entsprechend mit einem Icon gekennzeichnet.

Eine wichtige Änderung zu Release 7.50 ist, dass der AS ABAP nur noch als Unicode-System betrieben werden kann. Systeme mit Nicht-Unicode-Codepages werden nicht mehr unterstützt. Dementsprechend sind alle lauffähigen ABAP-Programme Unicode-Programme, und Nicht-Unicode-Programme werden nicht mehr unterstützt. Die Syntaxprüfung darf nur noch nach den Regeln der Unicode-Prüfung erfolgen. Die zugehörige PROGRAMMEIGENSCHAFT UNICODEPRÜFUNGEN AKTIV muss immer gesetzt sein.

Für eine genaue Historie der ABAP-Releases zwischen Release 7.02 und Release 7.50 sei auf die Webseiten <http://scn.sap.com/community/abap/blog/2013/05/22/abap-news-for-release-740> und <http://scn.sap.com/community/abap/blog/2015/10/20/abap-news-for-release-750-what-is-abap-750> verwiesen. Welche Neuerungen in welchem Release eingeführt wurden, lesen Sie unter ABAP – RELEASEABHÄNGIGE ÄNDERUNGEN der ABAP-Schlüsselwortdokumentation. Suchen Sie hier nach Begriff »News«.

Die ABAP-Schlüsselwortdokumentation bietet zu jedem Release den vollständigsten und aktuellsten Inhalt zur Programmiersprache ABAP. Sie umfasst neben der ABAP-Referenz unter anderem auch eine Referenzdokumentation zum ABAP Dictionary inklusive ABAP Core Data Services (CDS) sowie eine Referenz der Transformationssprache Simple Transformations (ST). Sie können die ABAP-Schlüsselwortdokumentation in aller Regel über die Funktionstaste **F1** in den Editoren der ABAP Workbench oder der ABAP Development Tools in Eclipse (ADT) aufrufen. Dort werden Onlineversionen angezeigt, die Ihnen eine gezielte Suche nach Begriffen und die direkte Ausführung von Beispielprogrammen der ABAP-Beispielbibliothek erlauben. Neben diesen backendbasierten Onlineversionen gibt es zusätzlich für jedes Release seit 7.02 auch statische Versionen im SAP Help Portal, wie z. B. unter [help.sap.com/abapdocu\\_750/de/index.htm](http://help.sap.com/abapdocu_750/de/index.htm) für das Release 7.50.

## Syntaxkonventionen in der ABAP-Referenz

Der ABAP-Referenz liegen die folgenden Syntaxkonventionen zugrunde:

- ▶ ABAP-Wörter werden in Großbuchstaben dargestellt.  
Beispiel:  
`APPEND, NON-UNIQUE, INTO`
- ▶ Operanden werden in Kleinbuchstaben dargestellt.  
Beispiel:  
`doobj, dbtab`

- ▶ Punkte, Kommata, Doppelpunkte und runde Klammern werden normal dargestellt. Sie sind Teil der ABAP-Syntax.

Beispiel:  
`., :, ,, ()`

- ▶ Operatoren werden normal dargestellt. Sie sind Teil der ABAP-Syntax.

Beispiel:  
`+, -, *, /`

- ▶ Teile von Anweisungen, die verwendet werden können, aber nicht müssen, sind in eckige Klammern `[ ]` gesetzt. Eine Aufzählung von Anweisungsteilen in eckigen Klammern bedeutet, dass alle oder einzelne Teile verwendet werden dürfen. Wenn mindestens ein Teil verwendet werden muss, ist dies im Text vermerkt. Eckige Klammern sind nicht Teil der ABAP-Syntax (Ausnahme: Tabellenausdrücke, Assoziationen in Mesh-Pfaden).

Beispiel:  
`[NO-GROUPING] [NO-SIGN] [NO-ZERO]`

- ▶ Vertikale Striche `|` zwischen Anweisungsteilen bedeuten, dass nur einer der aufgeführten Teile innerhalb einer Anweisung verwendet werden darf. Ein vertikaler Strich bindet immer die unmittelbaren Nachbarn. Striche sind nicht Teil der ABAP-Syntax. (Ausnahme: Begrenzer von Zeichenketten-Templates).

Beispiel:  
`DDMMYY | MMDDYY | YYMMDD`

- ▶ Geschweifte Klammern `{ }` fassen zusammengehörige Teile von Anweisungen zusammen, beispielsweise rechts oder links von vertikalen Strichen. Geschweifte Klammern sind nicht Teil der ABAP-Syntax (Ausnahme: eingebettete Ausdrücke in Zeichenketten-Templates).

Beispiel:  
`{NO EDIT MASK} | {EDIT MASK mask}`

- ▶ Punkte bedeuten, dass an dieser Stelle andere Teile der Anweisung stehen können.

Beispiel:  
`... AS CHECKBOX`

Die Syntaxdiagramme der ABAP-Referenz zeigen die Sprachelemente in einer syntaktisch korrekten Reihenfolge. In vielen Anweisungen sind auch andere Reihenfolgen möglich, werden aber nicht erwähnt. Für die Dokumentation wurde die Reihenfolge gewählt, die am besten zur Semantik einer Anweisung passt und mit der sich die verschiedenen Varianten einer Anweisung am besten einheitlich beschreiben lassen.

## Ausgaben von Quelltextbeispielen

Im Gegensatz zu früheren Auflagen der ABAP-Referenz wird in den Quelltextbeispielen weitgehend darauf verzichtet, Ausgaben mit der `WRITE`-Anweisung zu erzeugen. Die `WRITE`-Anweisung ist zwar vermeintlich einfach und bequem, beruht jedoch auf der klassischen Listentechnologie, die nicht mehr zu einer serviceorientierten Programmierung mit ABAP Objects passt,

wie sie heutzutage hauptsächlich umgesetzt wird. Stattdessen wird in den meisten Beispielprogrammen, die eine Ausgabe produzieren, die genau für diesen Zweck geschriebene Klasse `CL_DEMO_OUTPUT` verwendet. Die Klasse `CL_DEMO_OUTPUT` erlaubt es, mit `WRITE`-Methoden ABAP-Daten in einen XML-basierten Ausgabestrom zu schreiben und aus diesem eine HTML-Ausgabe mit der Methode `DISPLAY` auszugeben oder mit der Methode `GET` abzuholen. Sie steht ab Release 7.03/7.31 seit SP07 sowie in höheren Releases zu Verfügung. Zusätzlich wird auch die Klasse `CL_DEMO_INPUT` als Ersatz für die bislang verwendeten Selektionsbilder eingesetzt.

Die Klasse `CL_DEMO_OUTPUT` kann über statische Methoden und Instanzmethoden verwendet werden. Die folgenden Methoden erzeugen eine Ausgabe in einem Ausgabestrom:

- ▶ Die Methoden `BEGIN_SECTION`, `NEXT_SECTION` und `END_SECTION` erzeugen Überschriften (Header) und öffnen oder schließen Überschriftenebenen.
- ▶ Die Methoden `WRITE_DATA`, `WRITE_TEXT`, `WRITE_XML`, `WRITE_JSON` und `WRITE_HTML` schreiben verschiedene Ausgaben in den Ausgabestrom:
  - ▶ Mit der Methode `WRITE_DATA` können Sie elementare Datenobjekte (keine Referenzvariablen), Strukturen mit elementaren Komponenten und interne Tabellen solcher Zeilentypen schreiben.
  - ▶ Die anderen Methoden erzeugen formatierte Text-, XML-, JSON- oder HTML-Ausgaben.
- ▶ Die Methode `WRITE` ist generisch. Sie kann sowohl ABAP-Daten als auch Text (in einem nicht proportionalen Format) behandeln.
- ▶ Die Methoden `DISPLAY_...` (nur als statische Methoden verfügbar) funktionieren wie `WRITE_...`, schließen jedoch den aktuellen Ausgabestrom und öffnen einen neuen. Wenn SAP GUI verfügbar ist, wird die Ausgabe in einem SAP-GUI-Fenster angezeigt.
- ▶ Die Methode `LINE` erzeugt eine horizontale Linie.
- ▶ Die Methode `DISPLAY` schließt den aktuellen Ausgabestrom und öffnet einen neuen. Wenn das SAP GUI verfügbar ist, wird die Ausgabe in einem SAP-GUI-Fenster angezeigt. Optional können Sie auch Daten an die Methode `DISPLAY` übergeben, so wie es bei der Methode `WRITE` möglich ist.
- ▶ Die Methode `GET` funktioniert wie `DISPLAY`, aber zeigt die Daten nicht an. Stattdessen werden die formatierten Ausgabedaten in einem Textstring übergeben und können weiterverarbeitet werden.

Das Standardausgabeformat ist HTML. Optional können Sie ein einfaches Textformat mit Tabulatoren und Zeilenumbrüchen wählen. Das Format wählen Sie mit der Methode `SET_MODE` bei statischen Methoden bzw. über den Eingabeparameter `MODE` der Factory-Methode `NEW` bei Instanzmethoden.