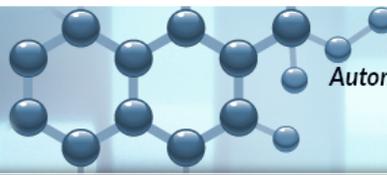


Das CT-Debug&Trace_Modul 4.0

Effektives Testen und Debuggen von ABAP™ Sourcen
Automatische Trace- Aufzeichnung, Backtrace-Funktionalität und
Variablenanalyse

SAP® Certified
Integration with SAP NetWeaver®

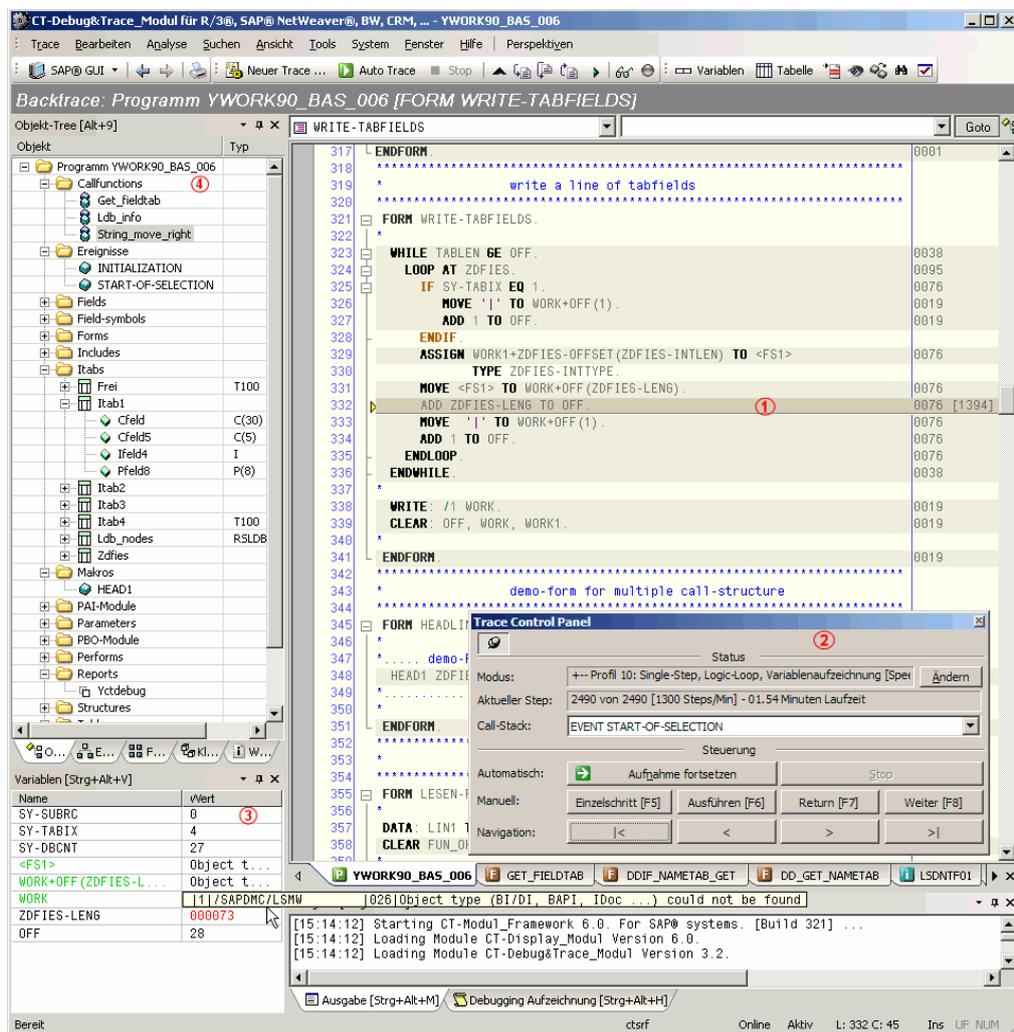


In Zukunft wird das Testen und Debuggen von ABAP® Programmen leichter, einfacher, effektiver ...

Aufwendiges Einzeldebugging das war gestern

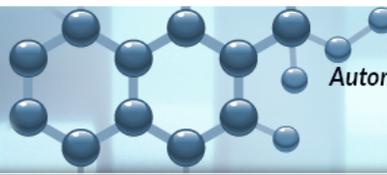
1. Die Aufzeichnung der prozessierten Befehle und Variablen ...

Das zeitraubende Einzelschrittverfahren beim traditionellen Debugging (F5, F6, F7 usw.) erledigt zukünftig Ihr Client-PC im online **automatisch**.



SCREEN Informationen

- ① Jede prozessierte Codingszeile wird im Source Code markiert.
- ② Das Steuerungspanel zeigt Ihnen LIVE den aktuellen Stand der gerade prozessierten Debug-Session.
- ③ Die prozessierten Variablen des aktuellen Befehls werden in diesem Bereich mit dem Feldinhalt gezeigt.
- ④ Der erweiterte Objekttree des prozessierten Programms dient der schnellen Doppelklicknavigation.



Sie können sich nun auf die **wirklich wichtigen Codingzeilen** konzentrieren. Überlassen Sie also die routinemäßigen Debuggingsschritte Ihrem PC, der ist schneller als Sie (viel schneller, ca. 1000 - 1500 Debuggingsteps/Min.).

2. Das manuelle Debugging mit automatischer Trace-Aufzeichnung ...

Sie können jederzeit das **schnelle automatische Debugging** anhalten, wie gewohnt **manuell** weiter debuggen und jederzeit das **Auto-Debugging wieder einschalten**. Weil sämtliche **maschinellen und manuellen** Einzelsteps aufgezeichnet werden, ist zu jedem Zeitpunkt eine Analyse der bisherigen Debuggschritte möglich. Zusätzlich wird in dem **aktuell** verwendeten Source Code jede prozessierte Zeile markiert.

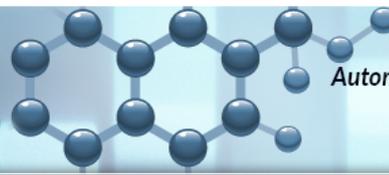
Wahlweise steht Ihnen auch das **synchronisierte** SE38-Debuggerdynpro (Klassischer ABAP® Debugger) an der jeweils aktuellen Codingstelle zur Verfügung ...

Zusätzlich wird auch der Ablauf der zur Laufzeit dynamisch generierten oder aufgerufenen Programme aufgezeichnet.

3. Der Backtrace-Modus - zurücksteppen auf dem Frontend-PC ...

Das CT-Debug&Trace_Modul stellt Ihnen auf dem Frontend-PC einen **Offline-Tracer** zur Verfügung. Sie können daher aufgrund der aufgezeichneten Tracedaten durch eine bereits **prozessierte Codingstelle** beliebig oft **vorwärts oder rückwärts** steppen und sehen gleichzeitig den betreffenden **Inhalt der prozessierten Variablen**.

Auch **während einer noch aktiven Debugging-Session** können Sie z.B. eine besonders relevante Codingstelle nochmals prüfen ...



Der Backtrace und der Inhalt der Variablen ...

The screenshot shows the CT-Debug & Trace_Modul 4.0 interface. The main window displays a backtrace for the program 'YWORK91_BAS_006 [FORM ERZEUGEN-PSEUDOZUFALL]'. The backtrace shows the following code:

```

135 DO LAENGE TIMES 0018
136
137 E_RAND = ( Z_A * E_RAND + 1 ) MOD Z_P. 0014
138 Z_ZUFALL = E_RAND / Z_P. 0014 [2361]
139 .....corresponding to the field length: Create a random number.....
140
141
142 IF TYP EQ 'N' 0014
143 DO LAENGE TIMES 0012
144 Z_ZUFALL = Z_ZUFALL * 10. 0006
145 ENDDO. 0006
146 MOVE Z_ZUFALL TO Z_AUSWAHL. 0006
147 Z_OFFSET = 29 - LAENGE + 1. 0006
148 ASSIGN Z_AUSWAHL+Z_OFFSET(1) TO <FS2>. "can be used directly 0006
149 IF P_TEST EQ 'J'. "only for test 0006
150 MOVE SY-INDEX TO ITAB2-LFIND.
151 MOVE E_RAND TO ITAB2-E_RAND.
    
```

A 'Trace-Liste für gewählte Zeile' popup window is open, showing the following table:

| Index | StepNr. | Variablen |
|-------|---------|---|
| 01 | 2213 | Z_ZUFALL= 0.0000000000..., Z_P= 9999999..., Z_A= 6713544... [...] |
| 02 | 2227 | Z_ZUFALL= 5.1778875707..., Z_P= 9999999..., Z_A= 6713544... [...] |
| 03 | 2241 | Z_ZUFALL= 1.9771160969..., Z_P= 9999999..., Z_A= 6713544... [...] |
| 04 | 2264 | Z_ZUFALL= 0.0000000000..., Z_P= 9999999..., Z_A= 6713544... [...] |
| 05 | 2278 | Z_ZUFALL= 4.1794204547..., Z_P= 9999999..., Z_A= 6713544... [...] |
| 06 | 2292 | Z_ZUFALL= 8.7239914698..., Z_P= 9999999..., Z_A= 6713544... [...] |
| 07 | 2306 | Z_ZUFALL= 8.9024122104..., Z_P= 9999999..., Z_A= 6713544... [...] |
| 08 | 2320 | Z_ZUFALL= 6.7379423258..., Z_P= 9999999..., Z_A= 6713544... [...] |
| 09 | 2343 | Z_ZUFALL= 0.0000000000..., Z_P= 9999999..., Z_A= 6713544... [...] |
| 10 | 2361 | Z_ZUFALL= 7.8112234740..., Z_P= 9999999..., Z_A= 6713544... [...] |
| 11 | 2388 | Z_ZUFALL= 0.0000000000..., Z_P= 9999999..., Z_A= 6713544... [...] |
| 12 | 2406 | Z_ZUFALL= 4.0691552612..., Z_P= 9999999..., Z_A= 6713544... [...] |
| 13 | 2424 | Z_ZUFALL= 8.4537399312..., Z_P= 9999999..., Z_A= 6713544... [...] |

The 'Variablen' window shows the following data:

| Name | Wert |
|----------|-----------------------------|
| SY-SUBRC | 0 |
| SY-TABIX | 1 |
| SY-DBCNT | 1 |
| E_RAND | 9.9412008730506778E+22 |
| Z_A | 671354420908421773035669 |
| Z_P | 999999999999999999999996467 |
| Z_ZUFALL | 7.8112234740524173E+01 |

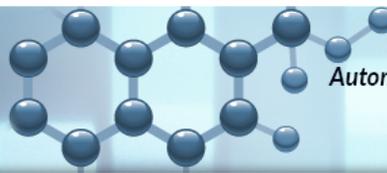
SCREEN Informationen

- ① Diese Coding-Zeile wurde 0014 mal prozessiert.
- ② Doppelklick im Coding auf '0014' zeigt Ihnen sämtliche 14 Stepnummern in einem PopUp.
- ③ Doppelklick im PopUp auf Stepnummer 2361 zeigt die Variablen des aktuellen Befehls für diesen Zeitpunkt (siehe ④).
- ⑤ Über die Navigationstasten '>' oder '<' können Sie z.B. ab Step 2361 im Einzelstep vorwärts/rückwärts das Programm analysieren.

4. Der Inhalt von Variablen wird automatisch ermittelt ...

Der Feldinhalt der an einem prozessierten Befehl beteiligten Variablen wird maschinell **vor und nach** der Befehlsdurchführung **ermittelt und aufgezeichnet** (max. 24 Variablen/Befehl).

Wenn Sie rationell die korrekte Programmausführung von einzelnen **FORMS**, **FUNCTIONS**, **METHODS** usw. prüfen wollen, erhalten Sie eine **automatische Registrierung** vom Inhalt der jeweiligen **Aufrufparameter** (Einzelfelder, Strukturen) **vor und nach** dem Aufruf.



5. Steuerung der Debugging-Session durch Debug-Profile ...

Durch **11 Debug-Profile** wird die Umsetzung Ihrer Debug-Strategie während der Debugging-Session wesentlich unterstützt. Sie steuern durch Auswahl eines geeigneten Ablauf-Profiles den **Umfang der Aufzeichnung**, die **Details** während des Programmablaufes und die **Dauer** der Debugging-Session.

Damit wird auch ein **umfangreicher Programmablauf** (z.B. Transaktion) **transparent** und die jeweils wichtigsten Ereignisse und Variablenergebnisse innerhalb des prozessierten Coding sind ersichtlich.

Wir zeigen Ihnen im Handbuch unter **'Tipps und Tricks'** wie Sie z.B. aus einer umfangreichen Transaktion gezielt eine bestimmte Debugging-Sequenz auswählen, prozessieren und die einzelnen Steps aufzeichnen (mit dem Inhalt der Variablen).

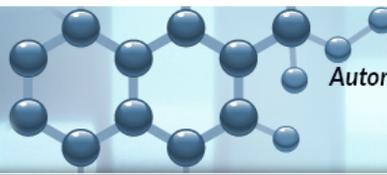
Umfangreiche Programme debuggen ...

The screenshot displays the CT-Debug & Trace_Modul 4.0 interface. On the left, the 'Workspace Tree' shows a project structure for 'YWORK90_BAS_006' with various function groups and modules. The main window shows a code editor with a trace window overlaid on the code. The trace window displays a list of steps with columns for 'STEPNO', 'STACK', and 'INDENTED EVENT'. A specific step (1849) is highlighted, showing a call to 'FORM SUBPROG-3'. Below the code editor, a 'Variablen' table is visible, listing variables and their values. The table includes variables like SY-SUBRC, SY-TABIX, SY-DBCNT, CFELD, NFELD, <FS1>, <FS2>, <GFS2>, CFELD+NFELD (IFELD), and IFELD.

| Name | Wert |
|---------------------|--|
| SY-SUBRC | 0 |
| SY-TABIX | 30 |
| SY-DBCNT | 1 |
| CFELD | ABCDEF6HIJKLMN0PQRSTUWV!*\$%1234 |
| NFELD | 00004 |
| <FS1> | No subproject with the name '&' exists for the project '&' |
| <FS2> | |
| <GFS2> | EF6HIJKLMN0PQRSTUWV!*\$%1234 0000 |
| CFELD+NFELD (IFELD) | EF6HIJKL |
| IFELD | 8 |

SCREEN Informationen

- ① Auch die Parameter von z.B. FORM/PERFORM werden aufgezeichnet (Step 1848).
- ② Der Parameterinhalt ist in einem breiten Ausgabefeld dargestellt.
- ③ In einer interaktiven CALL-Hierarchie ist die Position des FORM-Aufrufs (Step 1849) ersichtlich.
- ④ Die aufgerufenen Funktionsbausteine sind im Workspace-Tree dargestellt.
- ⑤ Die Entwicklungsklasse (Paket) zeigt zusätzliche Informationen.



6. Der Analysemodus mit 20 interaktiven Auswertungen ...

Zentraler Bestandteil der Tracedatenanalyse ist die **integrierte Darstellung** der prozessierten Programmteile (z.B. FORM, FUNCTION, METHOD ...) oder der einzelnen Debugging-Steps innerhalb eines **Debugging-Trees**.

Durch Doppelklick z.B. auf ein Tree-Element verzweigen Sie in die Detaildaten. Im **Analysemodus** werden Ihnen interaktiv mehr als **20 Standardauswertungen** angeboten, die jeweils auf die aktuelle prozessierte Codingstelle synchronisiert werden (Cockpit Information).

7. Art und Umfang der Dokumentation einer Debugging-Session ...

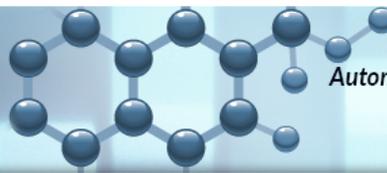
Während einer automatischen Debugging-Session von etwa **10 Minuten** werden z.B. im Einzelstep über die Steuerung eines Debug-Profiles mehr als **10.000 Verarbeitungsschritte** autonom aufgezeichnet. Zusätzlich werden etwa **20.000 Variablen** mit ihrem Inhalt registriert (ohne Steuerung oder Einflußnahme durch den User).

Umfang der Dokumentation ...

The screenshot displays the CT-Debug & Trace_Modul 4.0 interface. The main window shows the source code of a program named 'ERZEUGEN-PSEUDOZUFALL'. The code includes a loop 'DO LAENGE TIMES' and an 'IF' statement. The 'ZUFALL' variable is being assigned and used. The interface also shows a variable table on the right and a detailed trace table at the bottom.

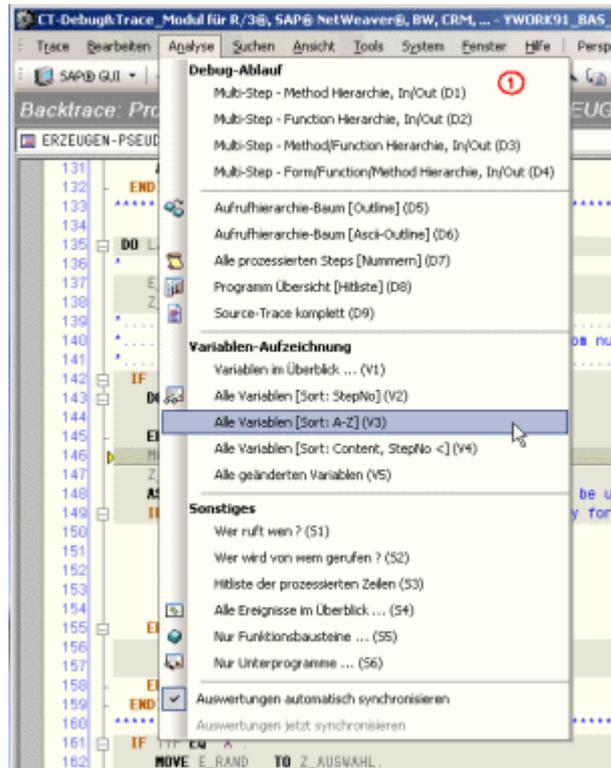
| NAME | WERT |
|-----------|--------------------------------|
| SY-SUBRC | 0 |
| SY-TABIX | 1 |
| SY-DBCNT | 1 |
| LAENGE | 2 |
| Z_ZUFALL | 5.3856137362226448E+01 |
| Z_AUSWAHL | 000000000000000000000000000044 |

| PHMMSS.MS | STEPNO | LINENO | C | P | VARIABLE_NAME | VARIABLE_CONTENT |
|-----------|---------|--------|---|-----------|--------------------------------|------------------|
| 121522967 | 0002365 | 000145 | I | SY-DBCNT | 1 | |
| 121522967 | 0002365 | 000145 | I | SY-SUBRC | 0 | |
| 121522967 | 0002365 | 000145 | I | SY-TABIX | 1 | |
| 121522967 | 0002365 | 000145 | X | Z_ZUFALL | 5.3856137362226448E+00 | |
| 121523030 | 0002366 | 000143 | I | LAENGE | 2 | |
| 121523030 | 0002366 | 000143 | I | SY-DBCNT | 1 | |
| 121523030 | 0002366 | 000143 | I | SY-SUBRC | 0 | |
| 121523030 | 0002366 | 000143 | I | SY-TABIX | 1 | |
| 121523264 | 0002367 | 000146 | P | LAENGE | 2 | |
| 121523264 | 0002367 | 000146 | I | SY-DBCNT | 1 | |
| 121523264 | 0002367 | 000146 | I | SY-SUBRC | 0 | |
| 121523264 | 0002367 | 000146 | I | SY-TABIX | 1 | |
| 121523264 | 0002367 | 000146 | I | Z_AUSWAHL | 000000000000000000000000000044 | |
| 121523264 | 0002367 | 000146 | X | Z_ZUFALL | 5.3856137362226448E+01 | |
| 121523374 | 0002368 | 000147 | I | LAENGE | 2 | |
| 121523374 | 0002368 | 000147 | I | SY-DBCNT | 1 | |
| 121523374 | 0002368 | 000147 | I | SY-SUBRC | 0 | |
| 121523374 | 0002368 | 000147 | I | SY-TABIX | 1 | |
| 121523374 | 0002368 | 000147 | X | Z_AUSWAHL | 000000000000000000000000000054 | |
| 121523374 | 0002368 | 000147 | I | Z_OFFSET | 29 | |



SCREEN Informationen

- ① Bereits während einer Debugging-Session stehen Ihnen bis zu 20 Auswertungen der aufgezeichneten Tracedaten zur Verfügung.
- ② In der Auswertung V3 können Sie z.B mit einem Blick erkennen, welcher Inhalt zu welchem Zeitpunkt in der Variablen 'Z_ZUFALL' enthalten ist.
- ③ Durch Doppelklick auf StepNo. 02367 in der Auswertung V3 erfolgt automatisch die Synchronisation auf den gleichen Zeitpunkt innerhalb der restlichen Auswertungen.



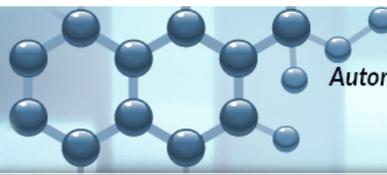
Darüber hinaus erfolgt zeitgleich automatisch die Übernahme der jeweils **prozessierten Codingmember** (z.B. bei einer umfangreichen Transaktion mehr als **100 Member**, mit etwa **30.000 Codingzeilen**) aus der Programmbibliothek des Applikationsservers zur **Analyse, Steuerungs- und Dokumentationszwecken**.

Die prozessierten **Sourcecodezeilen sind farbig markiert** und enthalten Informationen zur Navigation innerhalb der interaktiven Traceauswertungen.

8. Das intensive, gründliche Testen von Programmen ...

Durch das **Auto-Debugging** mit Steuerung durch Debug-Profile können Sie **maschinell** Ihr Programm prozessieren und in kurzer Zeit **umfassend testen**. Die automatische Aufzeichnung der einzelnen Debugging-Steps kann als **Testprotokoll** zusätzlich der technischen Programmdokumentation hinzugefügt werden.

Es ist daher auch möglich, zu einem späteren Zeitpunkt gezielt im **Offline-Modus** die für Sie wesentlichen Befehle in der Traceaufzeichnung in Ruhe **vorwärts oder rückwärts** zu steppen und **gleichzeitig** die beteiligten Variablen zu analysieren. Der jeweilige Inhalt der **aufgezeichneten prozessierten Variablen** ist über den gesamten Programmverlauf verfügbar.



9. Die Trennung von Aufzeichnung und Analyse im Debugging ...

Die schnelle maschinelle Aufzeichnung im Online-Debugging führt zu einem effizienteren Arbeitsablauf beim **Testen von Programmen** und der **Klärung von Fehlern**:

- Zunächst wird die Debugging-Session **aufgezeichnet** (im Hintergrund des PC),
- danach wird gezielt mit Unterstützung der **Auswertungsroutinen** eine **Analyse** vorgenommen.

Dabei ist zusätzlich zu berücksichtigen, dass auch die während der Originalaufzeichnung verwendeten **aktuellen Zugriffe auf Steuerungstabellen, Stamm- und Bewegungsdaten** implizit in der Dokumentation enthalten sind.

Die Wiederholung eines Testablaufes zu einem zurückliegenden Zeitpunkt, mit Rekonstruktion der jeweiligen Originaltabellen auf den Zeitpunkt des betreffenden Programmeinsatzes, dürfte vergleichsweise aufwendig sein.

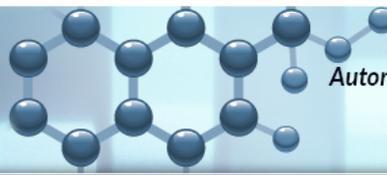
10. Weitere AddOn-Funktionen zur Unterstützung im Debugging ...

- **LIVE**-Informationen zum aktuellen Stand der Debug-Session durch **Trace Control Panel**,
- wahlfreie Positionierung zu **beliebigen** bereits prozessierten Befehlen,
- vereinfachten Darstellung von logischen Source Code Sequenzen durch **Outlinefunktionen**,
- **Tracepool** zur Rekonstruktion von Fehlersituationen, Testdaten, Prozeßabläufen,
- **Syntaxhighlighting** von von ABAP® Sourcen (Report, Programm, FuBau; Klassen ...),
- Objektbrowser (Programme, Klassen, Funktionsgruppen, Strukturen ...),
- Workspace Tree der **aktuell prozessierten** Source Code Member, Historie der Session,
- **Multi-Search-Funktionen** mit regulären Ausdrücken in gesonderten Ergebnis-Windows,
- Anzeige von **ToolTips** direkt an der Codingstelle (z.B. Deklarationsdaten von Variablen),
- individuelle Gestaltung der Programmoberflächen, **Default-Perspektiven** ...

11. Kommunikation mit SAP® R/3®, CRM®, BW®, Netweaver® ...

Sämtliche Module basieren auf unserem CT-Modul-Framework und sind für folgende SAP® Systeme geeignet: R/3® von Release 3.x bis einschließlich 4.7 (Enterprise), mySAP® ERP, mySAP® CRM, mySAP® BW, mySAP® APO und SAP® Netweaver® (bis 6.4).

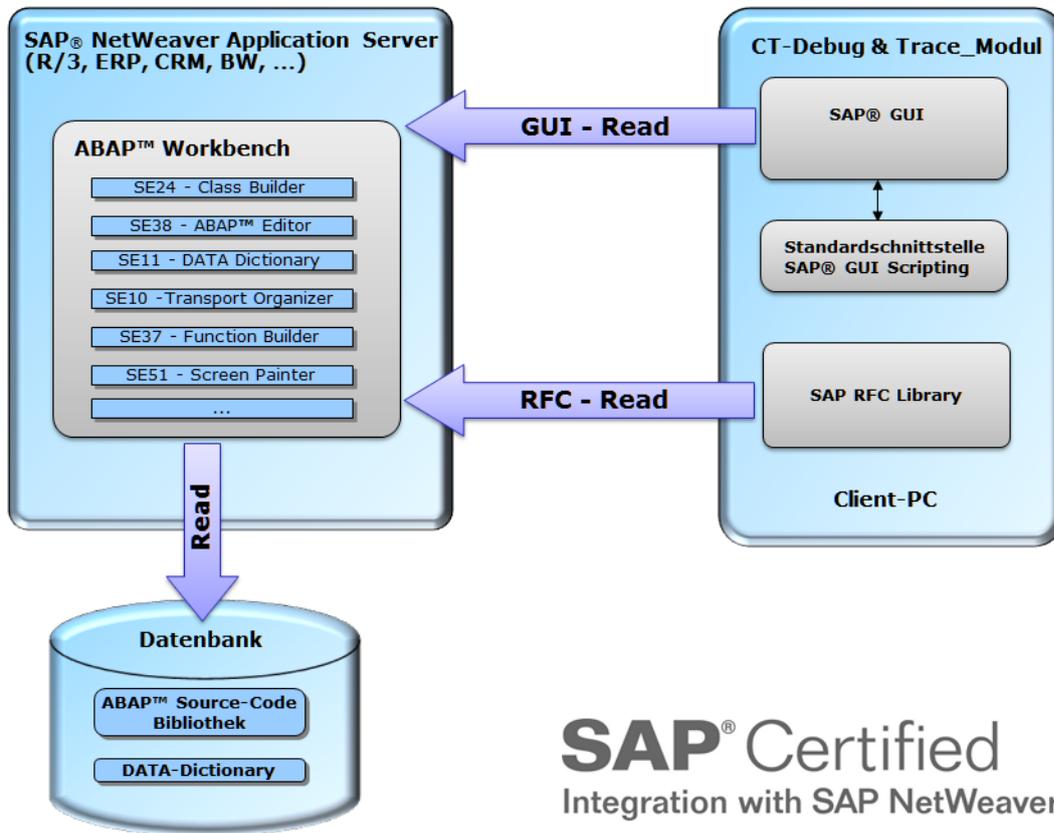
Die Add-On Entwicklungsmodule zeichnen sich durch eine zeitgemäße, ergonomische Oberfläche und effektive Arbeitsunterstützung aus, sie sind einfach zu installieren. Es sind **2** Installationsarten verfügbar:



Skizze

Kommunikation zwischen SAP® Applikationsserver und CT-Debug & Trace_Modul

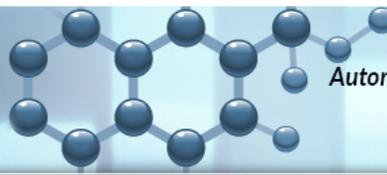
Das CT-Debug & Trace_Modul kommuniziert ausschließlich im Read-only mit dem Applikationsserver:



12. Fazit

Das effektive Testen und Debuggen von ABAP® Programmen wird durch den Einsatz des CT-Debug&Trace_Modul signifikant erleichtert.

Sie erhalten eine Testversion des CT-Debug&Trace_Modul unter:
www.ct-software.de/download.htm



13. Kontakt

CT-Softwareberatungs GmbH
Ziegeleiweg 8
33415 Verl
Germany

Web: www.ct-software.de
Email: sales@ct-software.de
Tel: +49-(0)5246-9310-15
Fax: +49-(0)5246-9310-16