

Inhalt

Geleitwort der SAP AG	17
Geleitwort der inconso AG	19
Einleitung	21

1 Einsatz des SAP Code Inspectors 29

1.1 Einordnung des SAP Code Inspectors	29
1.2 Verwendung des SAP Code Inspectors	34
1.2.1 Gemeinsamkeiten in verschiedenen Bereichen	35
1.2.2 Prüfvariante	38
1.2.3 Objektmenge	43
1.2.4 Inspektion	55
1.3 Überblick über die SAP-Standardprüfungen	63

2 Konfiguration und Funktionen des SAP Code Inspectors 67

2.1 Einstellmöglichkeiten des SAP Code Inspectors	67
2.1.1 Verwaltung von Tests	68
2.1.2 Verwaltung von Objektkollektoren	71
2.1.3 Verwaltung von Meldungsprioritäten	73
2.2 Objektkollektoren	75
2.2.1 Programme aus der Laufzeitanalyse	76
2.2.2 Verwendungsnachweis für Tabellen	78
2.2.3 Objekte aus Umfeldermittlung	79
2.2.4 Objekte aus Verwendungsnachweis	80
2.2.5 Objekte aus Coverage Analyzer	81
2.2.6 Objekte aus Datei-Upload	82
2.2.7 Objekte aus Laufzeitfehlern (ab Release 7.0 EHP2)....	83
2.2.8 Objekte aus cProjects (ab Release 7.0 EHP2)	84
2.2.9 Objekte aus eingebetteten Paketen (ab Release 7.0 EHP2)	85
2.2.10 Programme aus Katalog der Report-Sourcen (ab Release 7.0 EHP2)	86
2.3 Ergebnismeldungen unterdrücken	87
2.3.1 Pseudokommentare	88
2.3.2 Genehmigungsverfahren	95

2.4	Verwendungsnachweis für Prüfvarianten und Objektmengen	109
2.5	Ergebnisse von Inspektionen vergleichen	111
2.5.1	Vergleich zweier verschiedener Inspektionen	111
2.5.2	Vergleich zweier Versionen einer Inspektion	114
2.6	E-Mail versenden	117
2.6.1	E-Mail aus der Ergebnisliste einer Transaktion.....	117
2.6.2	E-Mail über den Report »rs_ci_email«	119
2.6.3	Fazit	123
2.7	Hintergrundjob des SAP Code Inspectors	123
2.7.1	Teilaufgabe Löschung	123
2.7.2	Teilaufgabe Import	124

3 Automatisierte Prüfungen mit dem SAP Code Inspector ... 125

3.1	Einsatzszenario für automatisierte Prüfungen	125
3.2	Inspektion als Job einplanen	126
3.3	Objektprüfungen bei Auftragsfreigabe	130
3.4	Objektprüfungen bei Aufgabenfreigabe	135
3.5	Externe Programmierschnittstelle des SAP Code Inspectors	150
3.5.1	Inspektion eines einzelnen TADIR-Objektes	151
3.5.2	Inspektion mit einer bestehenden Objektmenge	152
3.5.3	Inspektion von Sourcecode	152
3.5.4	Inspektion einer Selektion von Objekten	153
3.5.5	Inspektion einer Objektliste	154
3.6	Erfahrungen aus der Praxis	154
3.6.1	Planung eines automatisierten Code-Inspector-Einsatzes	155
3.6.2	Integration des SAP Code Inspectors in eigenen Code	157

4 Programmierung eigener Prüfungen für den SAP Code Inspector ... 159

4.1	Vorüberlegungen für eine eigene Prüfung	160
4.1.1	Generelle Planung	161
4.1.2	Datensammler des Code-Inspector-Frameworks.....	162
4.2	Grundlagen für eine eigene Prüfung	164
4.2.1	Hintergrund für die Grundlagen	164
4.2.2	Schritt 1: Erstellung von Einträgen im Prüfvariantenbaum	166

4.2.3	Kategorieneintrag erstellen	167
4.2.4	Prüfungseintrag erstellen	172
4.2.5	Schritt 2: Aktivierung der neuen Einträge	177
4.2.6	Aktivierte Einträge in der Ergebnisanzeige	178
4.3	Prüfvariante (Frontend)	179
4.3.1	Erweiterung des Prüfvariantenbaums	180
4.3.2	Parameterauswahlbildschirm	181
4.3.3	Methode »constructor«	187
4.3.4	Methode »if_ci_test~query_attributes«	193
4.3.5	Methode »cl_ci_query_attributes=>generic«	195
4.3.6	Struktur »sci_attent«	196
4.3.7	Feld »kind«	197
4.3.8	Methode »fill_messages«	200
4.3.9	Struktur »scimessage«	201
4.3.10	Methode »if_ci_test~navigate«	204
4.3.11	Methode »if_ci_test~display_documentation«	204
4.3.12	Methode »if_ci_test~exception«	204
4.4	Aufbau einer eigenen Prüfung (Backend)	205
4.4.1	Methode »run«	205
4.4.2	Methode »inform«	209
4.4.3	Methode »clear«	212
4.4.4	Methode »get_message_text«	212
4.4.5	Methode »run_begin«	212
4.4.6	Methode »run_end«	212
4.4.7	Methode »get_result_node«	213
4.4.8	Methode »consolidate_for_display«	213
4.4.9	Methode »modify_priorities«	214
4.4.10	Methode »add_obj_type«	215
4.4.11	Methode »get«	215
4.4.12	Sonstige Methoden	216
4.5	Tabellen der ABAP-Scan-Engine	216
4.5.1	Token-Tabelle	218
4.5.2	Statement-Tabelle	221
4.5.3	Structure-Tabelle	224
4.5.4	Level-Tabelle	226
4.6	Mögliche Umsetzungsszenarien	229
4.6.1	Manipulation bestehender SAP-Standardprüfungen	230
4.6.2	Integration eigener Prüfungen in den SAP Code Inspector	230
4.6.3	Fazit	234

5 Standardprüfungen des SAP Code Inspectors 235

5.1	Zuverlässigkeit und Relevanz von Prüfungen	237
5.1.1	Zuverlässigkeit	237
5.1.2	Laufzeit	238
5.2	Implizite Prüfungen des Code-Inspector-Frameworks	239
5.3	Allgemeine Prüfungen	240
5.3.1	Anweisungsstatistik	241
5.3.2	Tabellennamen aus SELECT-Anweisungen	243
5.3.3	Statistik der Tabelleneigenschaften	244
5.3.4	ABAP-Token-Statistik	245
5.4	Performanceprüfungen	246
5.4.1	Analyse der WHERE-Bedingung für SELECT	249
5.4.2	Analyse der WHERE-Bedingung für UPDATE und DELETE	254
5.4.3	SELECT-Anweisungen, die am Tabellenpuffer vorbei lesen	255
5.4.4	SELECT-Anweisungen mit anschließendem CHECK....	260
5.4.5	SELECT in Schleifen	262
5.4.6	Ändernde Datenbankzugriffe in Schleifen	263
5.4.7	Geschachtelte Schleifen	264
5.4.8	Kopieren großer Datenobjekte	265
5.4.9	Inperformante Operationen auf internen Tabellen....	266
5.4.10	Inperformante Parameterübergaben	269
5.4.11	Kopieren der aktuellen Tabellenzeile bei LOOP AT	274
5.4.12	'EXIT' oder keine Anweisung in SELECT-ENDSELECT-Schleife	276
5.4.13	Invalidierung des SAP-Tabellenpuffers	278
5.4.14	Verwendung von Indizes in der SELECT-Anweisung	280
5.4.15	Instanzerzeugung von BAdIs	282
5.4.16	SELECT INTO CORRESPONDING FIELDS bei gepufferten Tabellen	284
5.4.17	Prüfung der Tabelleneigenschaften	284
5.4.18	Performanceprüfungen, die es nicht gibt	294
5.5	Sicherheitsprüfungen	295
5.5.1	Kritische Anweisungen	297
5.5.2	Suche nach bestimmten kritischen Anweisungen (ab Release 7.0 EHP2)	300
5.5.3	Dynamische und mandantenabhängige Zugriffe im SELECT	301

5.5.4	Dynamische und mandantenabhängige Zugriffe mit INSERT, UPDATE, MODIFY, DELETE	303
5.5.5	Prüfung der SY-SUBRC-Behandlung	304
5.5.6	Verwendung der ADBC-Schnittstelle (ab Release 7.0 EHP2)	307
5.5.7	Ändernde Datenbankzugriffe außerhalb von Verbuchungsbausteinen	308
5.5.8	Mandantenabhängige Shared-Objects-Methoden (ab Release 7.0 EHP2)	309
5.5.9	Weitere Prüfmöglichkeiten zur Programmsicherheit	310
5.6	Syntaxprüfung/Generierung	310
5.6.1	Klassen/Interface-Konsistenz	311
5.6.2	Syntaxprüfung	312
5.6.3	Erweiterte Programmprüfung	313
5.6.4	Generieren von ABAP-Programmen	316
5.6.5	Suspekte Konvertierungen	317
5.7	Robuste Programmierung (ab Release 7.0 EHP2)	321
5.7.1	Suche nach APPEND und INSERT ... INDEX bei SORTED-Tabellen	322
5.7.2	Komplexe WHERE-Bedingung in SELECT-Anweisung	323
5.7.3	Prüfung der SY-SUBRC-Behandlung	326
5.7.4	Suspekte Konvertierungen	326
5.7.5	Anmerkungen zu SELECT ... FOR ALL ENTRIES und READ TABLE ... BINARY SEARCH	326
5.8	Programmierkonventionen	327
5.8.1	Namenskonventionen	328
5.8.2	Erweiterte Namenskonventionen für Programme	330
5.8.3	Testkonventionen von ABAP Unit	333
5.9	Metriken und Statistik	334
5.9.1	Metrik der ausführbaren Anweisungen	335
5.9.2	Prozedurale Metrik	337
5.9.3	Fan-out-strukturelle Metrik	341
5.9.4	Kommentarsprache-Metrik	342
5.9.5	OO-Größenmetrik	345
5.9.6	Anweisungsstatistik	347
5.9.7	ABAP-Web-Dynpro-Metrik (ab Release 7.0 EHP2)....	347
5.9.8	Worthäufigkeit in Kommentaren	350
5.10	Dynamische Tests	351
5.10.1	ABAP Unit	352

5.11	Oberflächen	355
5.11.1	GUI-Usability-Prüfung	355
5.11.2	Standardprüfungen für Web Dynpro	356
5.11.3	Web-Dynpro-Programmierkonventionen	358
5.11.4	Dynpro-Prüfungen	360
5.11.5	Dynpro-Generierung	360
5.11.6	Dynpro-Prüfung auf Usability und Accessibility	361
5.12	Suchfunktionen	362
5.12.1	Suche von ABAP-Token	363
5.12.2	Suche von ABAP-Anweisungsmustern	364
5.12.3	Suche nach unerwünschten Sprachelementen	365
5.12.4	Suche WRITE-Anweisungen	367
5.12.5	Suche Oracle Rule Hints	368
5.13	Anwendungsprüfungen	370
5.13.1	HR-Entkopplung: Prüfung der Infotypklassen	370
5.14	Interne Performancetests	371
5.14.1	Prüfung SQL-Trace: Analyse der WHERE-Bedingung für SELECT	373
5.14.2	Prüfung SQL-Trace: Analyse der WHERE-Bedingung für UPDATE und DELETE	375
5.14.3	Prüfung SQL-Trace: Zugriffe auf gepufferte Tabellen	376
5.14.4	Prüfung SQL-Trace: Explain für alle SELECT-Anweisungen	379
5.15	Interne Tests	380
5.15.1	Tests zu ENHANCEMENT-SECTION	381
5.15.2	Test zu CL_ABAP_COMPILER	382
5.15.3	Erkennen von totem Coding	382
5.15.4	Leerer Test	383
5.15.5	Überprüfung der Erweiterbarkeit von Tabellen	383
5.16	Proxy-Prüfungen	384
5.16.1	Proxy-Prüfungen	385

Anhang

389

A	Konstanten des SAP Code Inspectors	391
A.1	Inspektionsverarbeitung	391
A.2	Pseudokommentare und Genehmigungsverfahren	393
A.3	ABAP-Scan-Engine	397

B Meldungen der SAP-Standardprüfungen	405
B.1 Allgemeine Prüfungen	407
B.2 Performanceprüfungen	410
B.3 Sicherheitsprüfungen	418
B.4 Syntaxprüfung/Generierung	421
B.5 Programmierkonventionen	423
B.6 Metrik und Statistik	428
B.7 Dynamische Tests	431
B.8 Oberflächen	435
B.9 Suchfunktionen	440
B.10 Anwendungsprüfungen	441
B.11 Interne Performancetests	442
B.12 Interne Tests	444
B.13 Proxy-Prüfungen	446
B.14 Liste der internen Prüfungen	446
C Glossar	449
D Die Autoren	453
Index.....	455