

ABAP in Eclipse – Overview

Disclaimer

This presentation outlines our general product direction and should not be relied on in making a purchase decision. This presentation is not subject to your license agreement or any other agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or to develop or release any functionality mentioned in this presentation. This presentation and SAP's strategy and possible future developments are subject to change and may be changed by SAP at any time for any reason without notice. This document is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP assumes no responsibility for errors or omissions in this document, except if such damages were caused by SAP intentionally or grossly negligent.

Eclipse

Basics

The screenshot shows the Eclipse IDE interface with several components highlighted by yellow arrows and labels:

- Menus:** The top menu bar with options like File, Edit, Source, Navigate, Search, Project, Run, Window, and Help.
- Toolbar:** The toolbar below the menu bar containing various icons for file operations and development actions.
- Workbench window:** The main workspace area containing multiple views.
- Views:** The Project Explorer on the left showing a project structure with folders like 'Source Library' and 'Classes', and the Outline view below it showing a list of methods for the selected class.
- Editor:** The central editor window displaying the source code of the class `ZCL_EPM_DATA_ACCESS`.
- Perspectives:** The bottom-right area showing the Properties view for the selected class, with tabs for General and Source.
- Status line:** The bottom-most bar showing the current state of the editor, including 'Read-Only', 'Smart Insert', and the current file path.



Working with ABAP Projects

Creating ABAP Projects

Connecting to ABAP Backend system

The image shows two screenshots of the SAP ABAP Development Tools interface. The left screenshot shows the 'New' menu with 'ABAP Project...' selected. The right screenshot shows the 'New ABAP Project' dialog box with the 'System Connection' tab active, displaying a list of SAP systems. Annotations with callouts provide step-by-step instructions.

1. Right-click → New → ABAP Project...

2. press control + space to bring up the list of ABAP systems

ABAP systems configured in SAP Logon pad

System Connection
Select a connection to an SAP system. You can maintain system connections in SAP Logon

Connection:	Description:
CSS [PUBLIC] - CSS - Customer Service System - DON'T DELETE	
H74 [10.66.181.104]	
ISP [PUBLIC] - ISP ERP2005 HANA -Productive System-(MSCS C	
UI3 [PUBLIC] - 7.40 SAP_BASIS System	
UI4 [PUBLIC] - NW 7.40 SAP_BASIS Testsystem on HANA	
UIA [PUBLIC] - NW 7.41 HDB Content Development	
Y3S [PUBLIC] - NW 731 SAP_BASIS correction system	

7 results

UIA [PUBLIC]
NW 7.41 HDB Content Development

System ID: UIA
Connection Type: Load Balanci
Message Server: Idciuia.wdf.sap
Group: PUBLIC

Secure Network Communicatio
SSO enabled: No

< Back Next > Finish Cancel

Creating ABAP Projects

Connecting to ABAP Backend system

New ABAP Project

Logon to System

Specify a value for field 'Password'

System ID: UI4

Client: * 000

User: * SANDU

Password: *

Change Default Language

Language: * EN

3. Enter user logon details

Password disabled if SSO is enabled for system

< Back Next > Finish Cancel

New ABAP Project

Project Name and Favorite Packages

Enter a project name and select a set of favorite packages for the new ABAP project

Project Name: UIA_000_sandu_en_1

Favorites Packages:

Add... Remove

4. Add favorite packages

Working sets

Add project to working sets

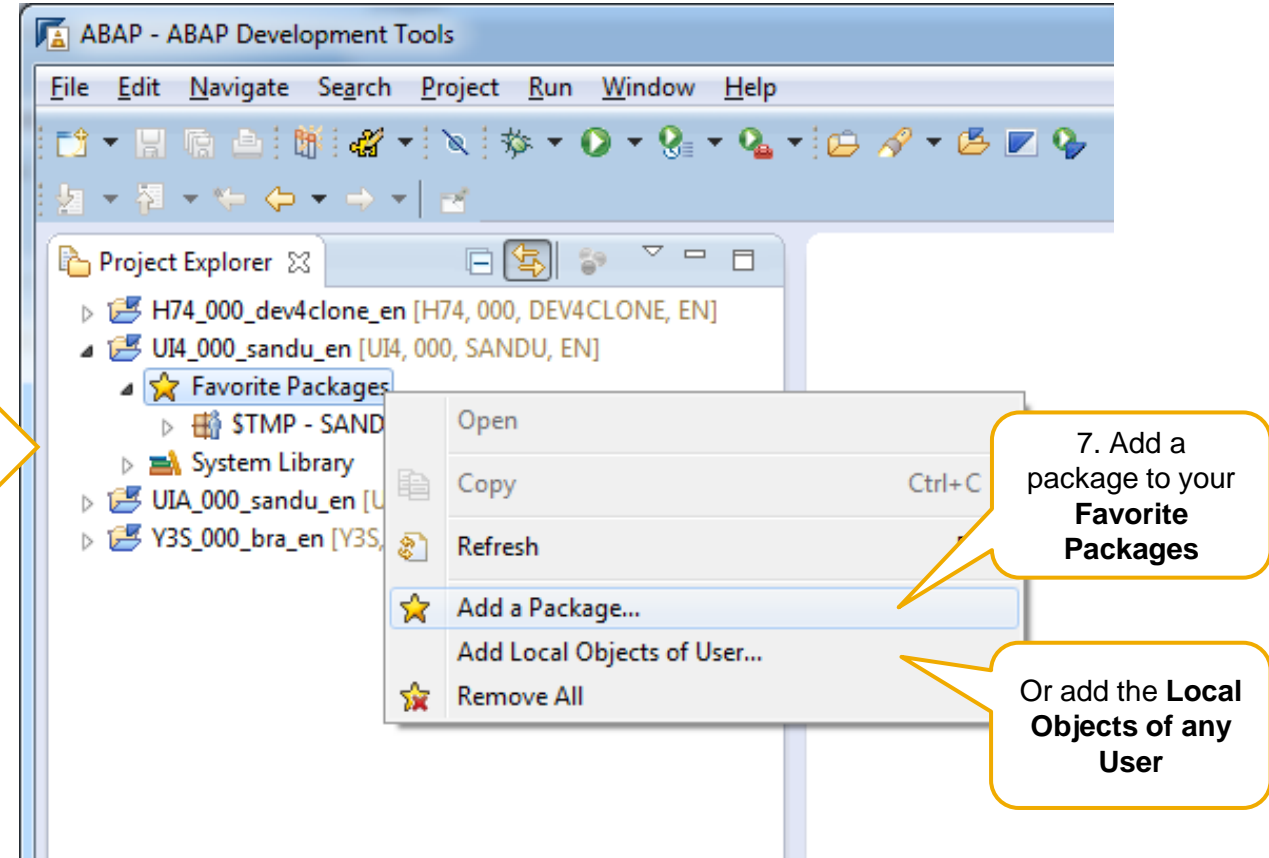
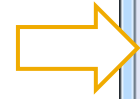
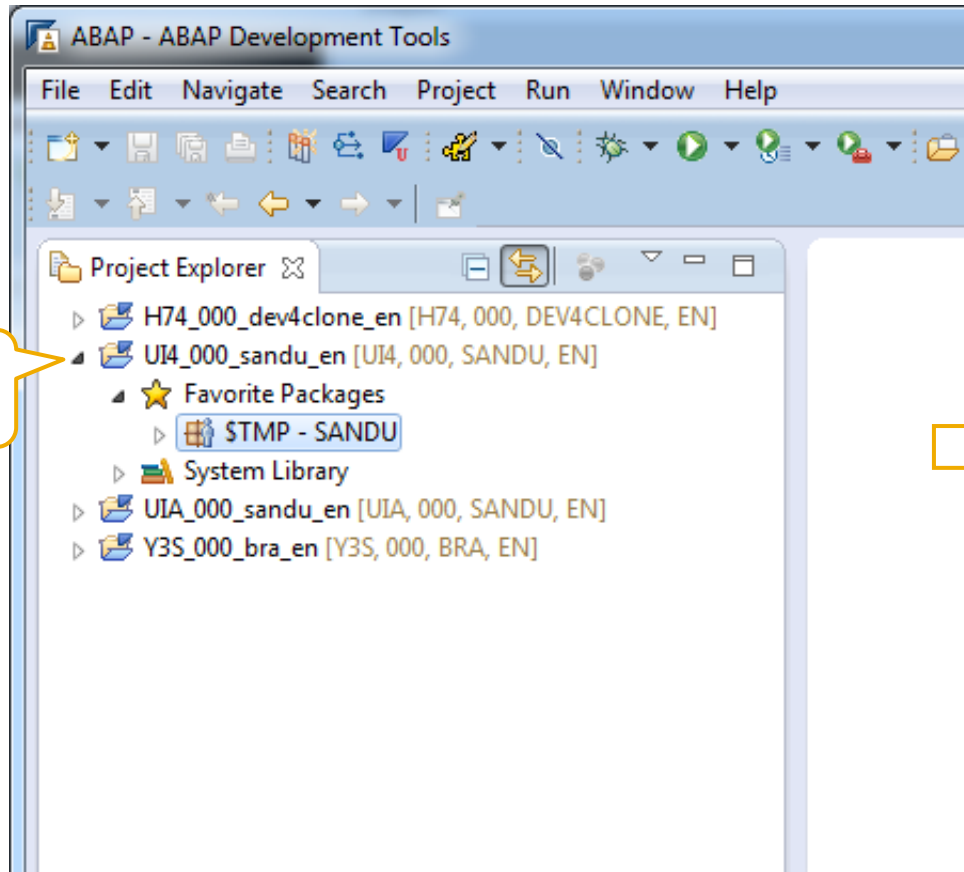
Working sets: Select...

5. Click on Finish

< Back Next > Finish Cancel

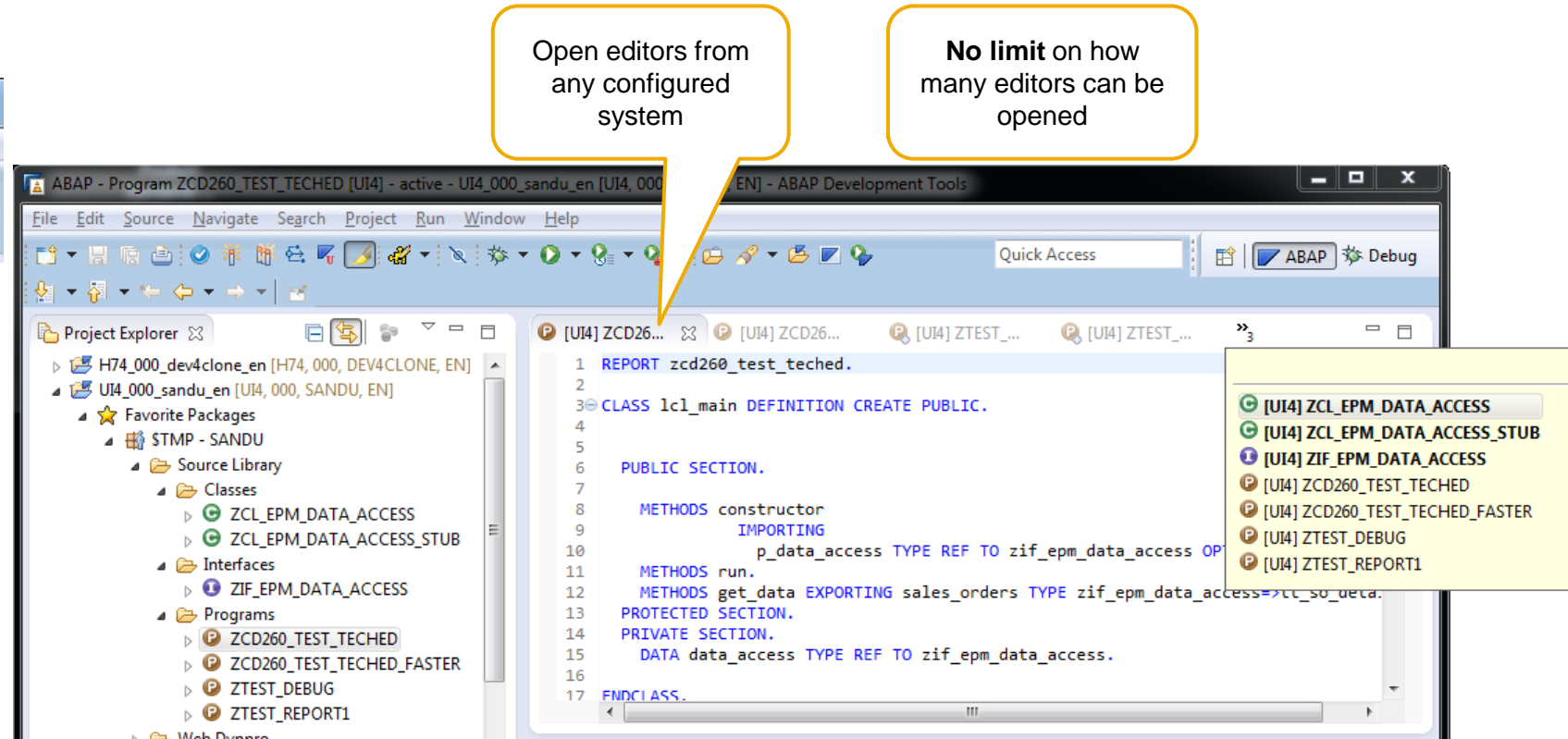
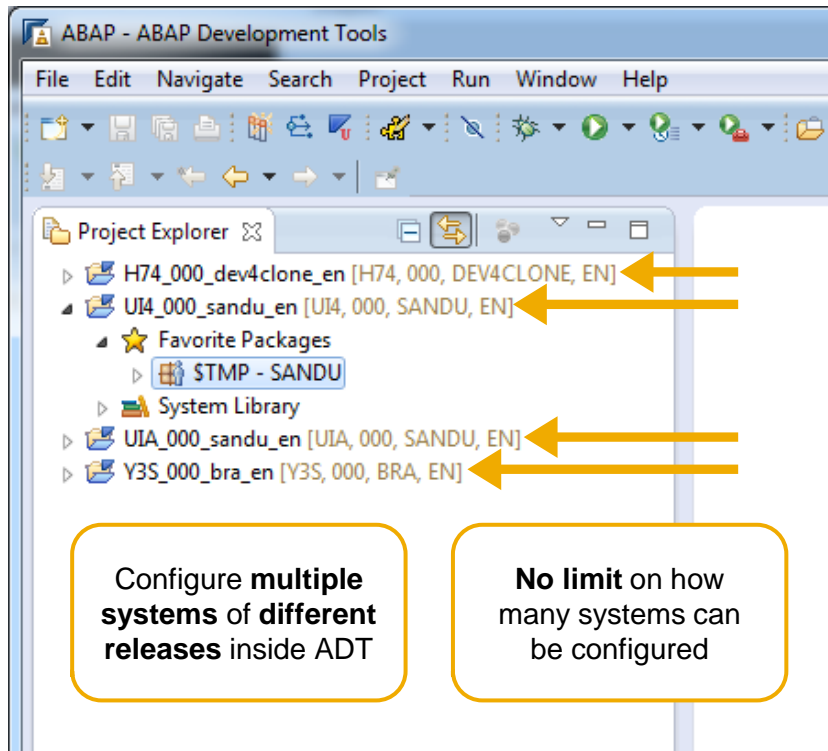
Creating ABAP Projects

Adding or Removing Favorite Package



Creating ABAP Projects

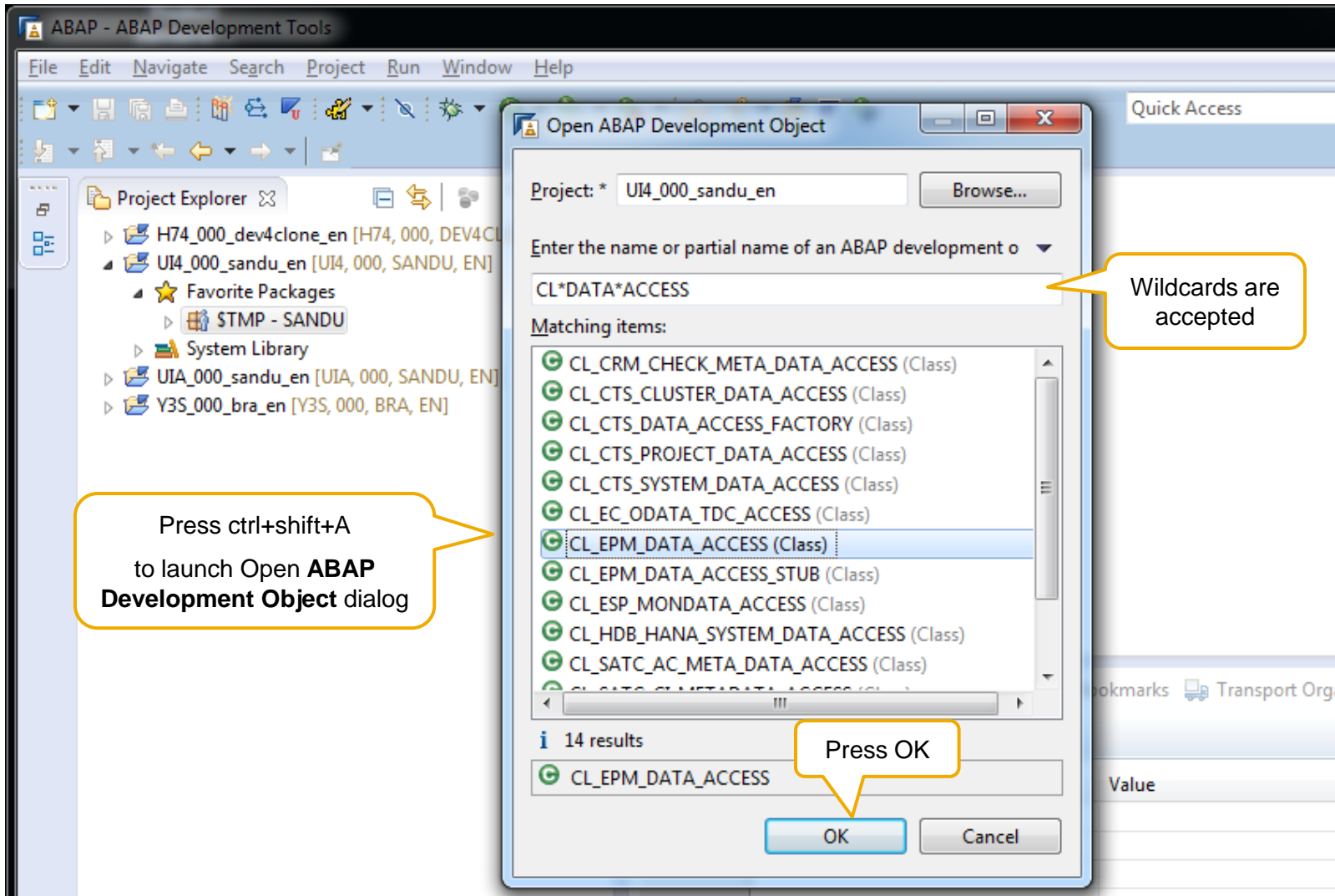
Configuring multiple backend systems





Working with Development Objects

Opening ABAP Development Objects



Link with Editor

1. ABAP Object's editor is opened

```
1  "!" Productive implementation of <em>IF_EPM_DATA_ACCESS</em>. Accesses tables or  
2  CLASS cl_epm_data_access DEFINITION  
3  PUBLIC  
4  FINAL  
5  CREATE PUBLIC .  
6  
7  PUBLIC SECTION.  
8  
9  
10  
11  
12  
13  
14  
15
```

2. Click on Link With Editor...

3. ...to open the containing package in project explorer

```
1  "!" Productive implementation of <em>IF_EPM_DATA_ACCESS</em>. Accesses table  
2  CLASS cl_epm_data_access DEFINITION  
3  PUBLIC  
4  FINAL  
5  CREATE PUBLIC .  
6  
7  PUBLIC SECTION.  
8  INTERFACES if_epm_data_access.  
9  PROTECTED SECTION.  
10 PRIVATE SECTION.  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  

```

Opening Development Objects in the SAP GUI

SAP GUI Integration

1. Press ctrl+shift+A, to launch Open **ABAP Development Object** dialog

2. Enter name of DDIC ABAP object

3. DDIC objects don't have Eclipse native editors. And SAP GUI editor opens

SAP GUI editor is embedded inside ADT

The screenshot shows the SAP GUI interface with the 'Open ABAP Development Object' dialog box open. The dialog box is titled 'Project: * UI4_000_sandu_en' and contains a search field with 'SFLIGHT' entered. Below the search field, a list of matching items is displayed, including 'SFLIGHT (Database Table)', 'SFLIGHT (View Entity)', 'SFLIGHT (Business Object Type)', 'SFLIGHT (Search Help)', 'SFLIGHTS (View)', 'SFLIGHTS2 (View)', 'SFLIGHT_ADT_CARRIERS (Table Type)', 'SFLIGHT_ADT_ST_FLIGHT (Transformation)', 'SFLIGHT_ADT_ST_FLIGHTS (Transformation)', and 'SFLIGHT_AND_SPFLI_VIEW (View Entity)'. The 'SFLIGHT (Database Table)' item is selected. An arrow points from this dialog to the main SAP GUI window. The main window shows the 'Project Explorer' on the left, which is expanded to show the 'Dictionary' folder. The 'SFLIGHT' object is selected in the 'Database Tables' folder. The main window also shows the 'Dictionary: Display Table' editor, which displays the structure of the 'SFLIGHT' table. The table structure is as follows:

Field	Key	Ini...	Data element	Data Type	Length	Deci...	Short Description
MANDI	✓	✓	S_MANDI	CLNT	3	0	Client
CARRID	✓	✓	S_CARR_ID	CHAR	3	0	Airline Code
CONNID	✓	✓	S_CONN_ID	NUMC	4	0	Flight Connection N
FLDATE	✓	✓	S_DATE	DATS	8	0	Flight date
PRICE	✓	✓	S_PRICE	CURR	15	2	Airfare
CURRENCY	✓	✓	S_CURR_CODE	CUKY	5	0	Local currency of air
PLANETYPE	✓	✓	S_PLANETYPE	CHAR	10	0	Aircraft Type

Opening TCodes in the SAP GUI

SAP GUI Integration

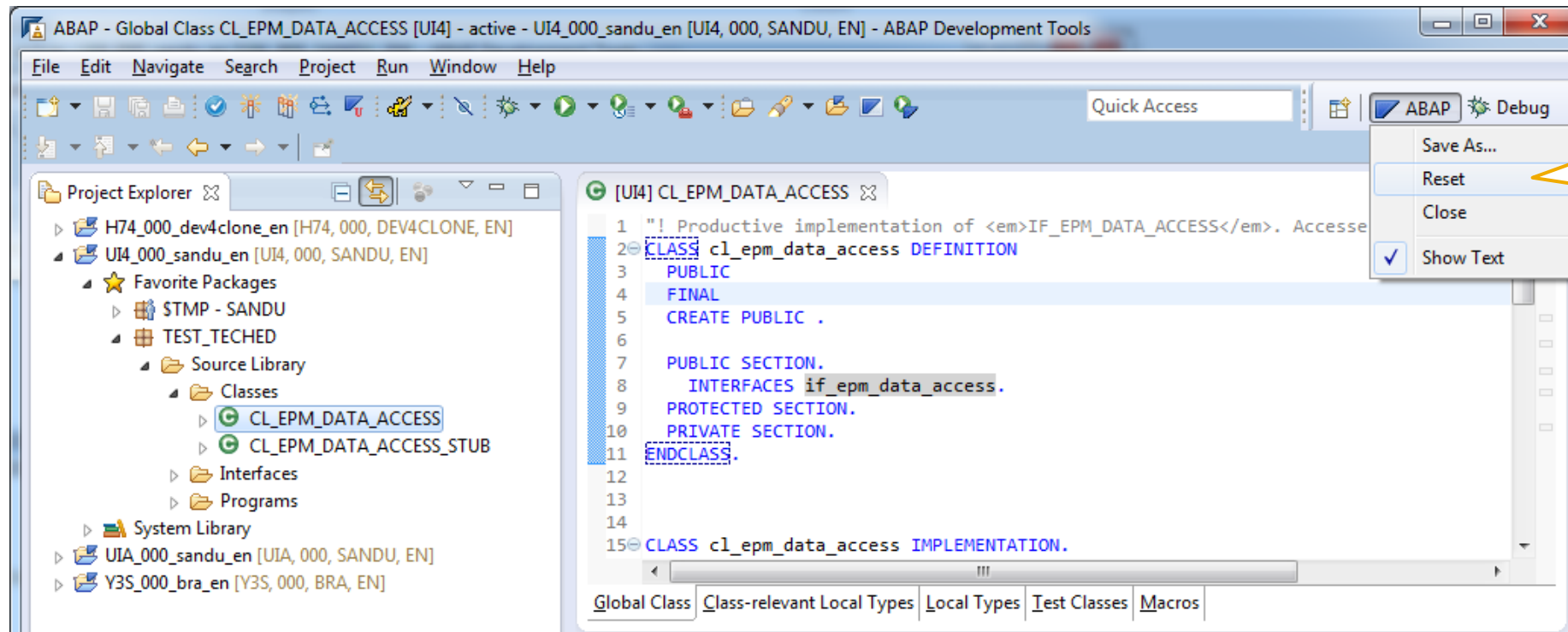
1. To open any **SAP GUI transaction**, click on this toolbar button...

2. ...and SAP GUI transaction editor is opened

3. Type in any **transaction**

Field	Key	Ini...	Data element
MANDT	✓	✓	S_MANDT
CARRID	✓	✓	S_CARR_ID
CONNID	✓	✓	S_CONN_ID
FLDATE	✓	✓	S_DATE
PRICE	□	✓	S_PRICE
CURRENCY	□	✓	S_CURRCODE

Resetting the perspective



Click on Reset to restore the ABAP perspective to default layout

Helps when you accidentally close a view or when you made lot of changes to the layout



Editing ABAP Source Code

Basics

ABAP Source Code Editor

If lock is taken, an indicator is shown

If ABAP class is being edited, then a * is shown

The system containing the ABAP Class is shown

New ABAP Class editor in ADT.

It's a full source based editor. i.e. the complete source code of the ABAP class is shown

```
*[UI4] CL_EPM_DATA_ACCESS ⓘ  
1  "! Productive implementation of <em>IF_EPM_DATA_ACCESS</em>. Accesses tables or  
2  CLASS cl_epm_data_access DEFINITION  
3    PUBLIC  
4    FINAL  
5    CREATE PUBLIC .  
6  
7    PUBLIC SECTION.  
8      INTERFACES if_epm_data_access.  
9    PROTECTED SECTION.  
10   PRIVATE SECTION.  
11  ENDClass.  
12  
13  
14  
15  CLASS cl_epm_data_access IMPLEMENTATION.  
16
```

ABAP language keywords are show in blue

Getting Support from the Content Assist

ABAP Source Code Editor

The screenshot shows the ABAP Source Code Editor with the following code:

```
37 "add error handling
38 ENDMETHOD.
39
40 alv_table->display( ).
41 ENDMETHOD.
42
43
44 METHOD get_data.
45 DATA:
46 lt_sales_order TYPE STANDARD TABLE OF snwd_so,
47 ls_so TYPE if_epm_data_access=>ty_so_details,
48 ls_so_item TYPE snwd_so_i,
49 ls_pd TYPE snwd_pd.
50
51
52 me->data_access->select_sales_orders(
53 EXPORTING
54 max_rows = 100
55 IMPORTING
56 et_sales_orders = lt_sales_order ).
57
58 LOOP AT lt_sales_order INTO DATA(ls_sales
59
60 CLEAR ls_so.
61 ls_so-so_id = ls_sales_order-so_id.
62
```

Annotations and callouts:

- 1. Press ctrl+space anywhere in the editor to bring up the content assist
- Shows more info on the selected entry "tt_sales_order"
- Press 'Shift+Enter' to insert full signature
- 2. Press Shift+Enter to insert the complete signature

Working with Source Code Templates

Code Templates and Custom Code Templates

1. Drag and drop template into the ABAP editor to insert the code snippet

```
59 EXPORTING
60 text TYPE snwd_text-text.
61
62 "!" Reads sales order headers including item informa
63 "!" This method allows to combine all steps don in
64 "!" <em>SELECT_SALES_ORDER_ITEMS</em>, <em>SELECT_PP
65 METHODS
66 select_sales_order_details
67 IMPORTING
68 max_rows TYPE i DEFAULT 1000
69 EXPORTING
70 so_details TYPE tt_so_details.
71
```

Predefined code templates from "Templates" View

Name	Description
assertEquals	assert_equals (ABAP Unit)
case	Case block
catchAll	##CATCH_ALL pragma (unhandled exception o...
debuggerTrace	Trace requests for ABAP debugger

2. Open Window → Preferences

3. Configure existing code templates...

4. Or define your own code template

Name	Context	Description	Auto Ins...
<input checked="" type="checkbox"/> assertEquals	ABAP	assert_equals (ABAP Unit)	on
<input checked="" type="checkbox"/> case	ABAP	Case block	on
<input checked="" type="checkbox"/> catchAll	ABAP	##CATCH_ALL pragma (u...	on
<input checked="" type="checkbox"/> debuggerTr...	ABAP	Trace requests for ABAP de...	on
<input checked="" type="checkbox"/> firePlug	Web Dynpro ABAP	Fire Plug (use in view contr...	on
<input checked="" type="checkbox"/> generateMe...	Web Dynpro ABAP	Generate Message Via Mes...	on
<input checked="" type="checkbox"/> if	ABAP	If block	on
<input checked="" type="checkbox"/> ifElse	ABAP	If-else block	on
<input checked="" type="checkbox"/> instantiateU...	Web Dynpro ABAP	Instantiated Used Compon...	on
<input checked="" type="checkbox"/> lcl	ABAP	Local class	on
<input checked="" type="checkbox"/> lif	ABAP	Local interface	on

Frequently used functions

ABAP Source Code Editor

1. Right-click anywhere inside source code editor

```
42  
43  
44 METHOD get_data.  
45 DATA:  
46   lt_sales_order  
47   ls_so  
48   ls_so_item  
49   ls_pd  
50  
51  
52   me->data_access  
53   EXPORTING  
54     max_rows  
55   IMPORTING  
56     et_sales_or  
57  
58 LOOP AT lt_sale  
59  
60   CLEAR ls_so.  
61   ls_so-so_id =  
62  
63   me->data_acce  
64   EXPORTING  
65     so_guid  
66   IMPORTING
```

- Undo (Ctrl+Z)
- Revert File
- Save (Ctrl+S)
- Navigate To (F3)
- Open Declaration (Shift+F3)
- Quick Type Hierarchy (Ctrl+T)
- Open ABAP Type Hierarchy (F4)
- Open in Project
- Open With
- Open Others
- Show In (Alt+Shift+W)
- Quick Outline (Ctrl+O)
- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Copy ABAP Resource URL
- Paste (Ctrl+V)
- Get Where-used List... (Ctrl+Shift+G)
- Search Content...
- Rename (Alt+Shift+R)
- Quick Fix (Ctrl+1)
- Source (Alt+Shift+S)
- Profile As
- Debug As
- Run As
- Coverage As
- Validate
- Team
- Compare With
- Replace With
- Unlock (Ctrl+U)
- Preferences...
- Remove from Context (Ctrl+Alt+Shift+Down)

- Toggle Comment (Ctrl+7)
- Add Comment (Ctrl+<)
- Remove Comment (Ctrl+>)
- Rename (Alt+Shift+R)
- Extract Method (Alt+Shift+M)
- Format (Shift+F1)
- Format Block (Ctrl+Shift+F1)
- Delete unused variables (all) (Alt+U)
- Delete unused variables (selection) (Alt+Shift+U)
- Occurrences in File (Ctrl+Shift+U)
- Show Code Element Information (F2)
- Show Method Information (Alt+F2)

2. Many useful and frequently used functions available here



Problems, bookmarks and other markers

Checking ABAP Syntax

Automatic Syntax Checks

Automatic Syntax check is enabled for ABAP Class\Interface\Report s\Function Modules

Opening the class automatically will show all the errors

```
[Y3S] ZCD260_BR_TEST_TECHED
```

```
30 METHOD run.  
31   get_data(  
32     IMPORTING  
33     sales_orders = DATA(sales_orders) ).  
34 TRY.  
35   cl_salv_table=>factory( IMPORTING r_salv_table = DATA(alv_table) CHANGING  
36     CATCH cx_salv_msg.  
37     "add error handling  
38   ENDTRY.  
39  
40   alv_table->display( ).  
41 ENDMETHOD.  
42  
43  
44 METHOD get_data.  
45   DATA:  
46     lt_sales_order TYPE STANDARD TABLE OF snwd_so,  
47     ls_so           TYPE zif_epm_data_access=>ty_so_details,  
48     ls_so_item     TYPE snwd_so_i,  
49     ls_pd          TYPE snwd_pd.  
50  
51
```

No need to press ctrl+F2 anymore

Marking Occurrences in the Source Code Editor

The screenshot shows the SAP Source Code Editor with the following code:

```
92  
93 ENDCLASS.  
94  
95 START-OF-SELECTION.  
96  
97 NEW lcl_main( )->run( ).  
98  
99 "! unit test for local lcl_main class using the stub implementation to make  
100 SS ltcl_unit_test DEFINITION FINAL FOR TESTING  
101 DURATION SHORT  
102 TASK LEVEL HARMLESS.  
103  
104 PRIVATE SECTION.  
105 DATA sut TYPE REF TO if_epm_data_access.  
106 METHODS:  
107 first_test FOR TESTING RAISING cx_static_check.  
108 ENDCLASS.  
109  
110  
111 CLASS ltcl_unit_test IMPLEMENTATION.  
112  
113 METHOD first_test.  
114  
115  
116 CREATE OBJECT sut TYPE cl_epm_data_access_stub.  
117 NEW lcl_main( sut )->get_data( IMPORTING sales_orders = DATA(sales_orders)  
118 cl_abap_unit_assert=>assert_equals( exp = 4 act = lines( sales_orders ) ).  
119 ENDMETHOD.  
120  
121 ENDCLASS.
```

Annotations in the image:

- 1.** Select any variable and all its occurrences are highlighted: A yellow callout points to the variable `sut` on line 105, which is highlighted in blue.
- 2.** A yellow callout points to the variable `sut` on line 116, which is highlighted in blue.
- 3.** A yellow callout points to the variable `sut` on line 117, which is highlighted in blue.
- 4.** The occurrences are also highlighted here: A yellow callout points to the variable `sut` on line 118, which is highlighted in blue.

Working with Bookmarks

The screenshot illustrates the steps to add and use bookmarks in the SAP IDE. On the left, the Project Explorer shows a context menu for the object 'UIA_000_sandu_en'. A callout box points to the 'Add Bookmark' option, with the instruction: "1. Add bookmark for any object".

The main editor window shows the source code of the method '[UI4] IF_EPM_DATA_ACCESS'. A callout box points to a small bookmark icon (a vertical line) on the left margin of the code, with the instruction: "2. Right-click and choose 'Add bookmark'".

A second callout box points to the bookmark icon in the Bookmarks view at the bottom right, with the instruction: "Bookmark icon".

The Bookmarks view displays a table of 5 items:

Description	Resource	Path	Location
automatic syntax checks	zcd260_br_tes...	/Y3S_000_bra_en/.a...	Unknown
TODO in method select_text	if_epm_data_...	/UI4_000_sandu_en...	line 55
WDA complex context mapping	v_assign_step...	/UIA_000_sandu_en...	Unknown
WDA context mapping	uielementpro...	/UIA_000_sandu_en...	Unknown
WDA View designer	viewbutton.w...	/UIA_000_sandu_en...	Unknown

A callout box points to the 'TODO in method select_text' entry, with the instruction: "3. Double click on the bookmark to open the respective editor and navigate to the line."

ABAP Resource URL of Development Objects

Configuring use of ABAP Resource URL

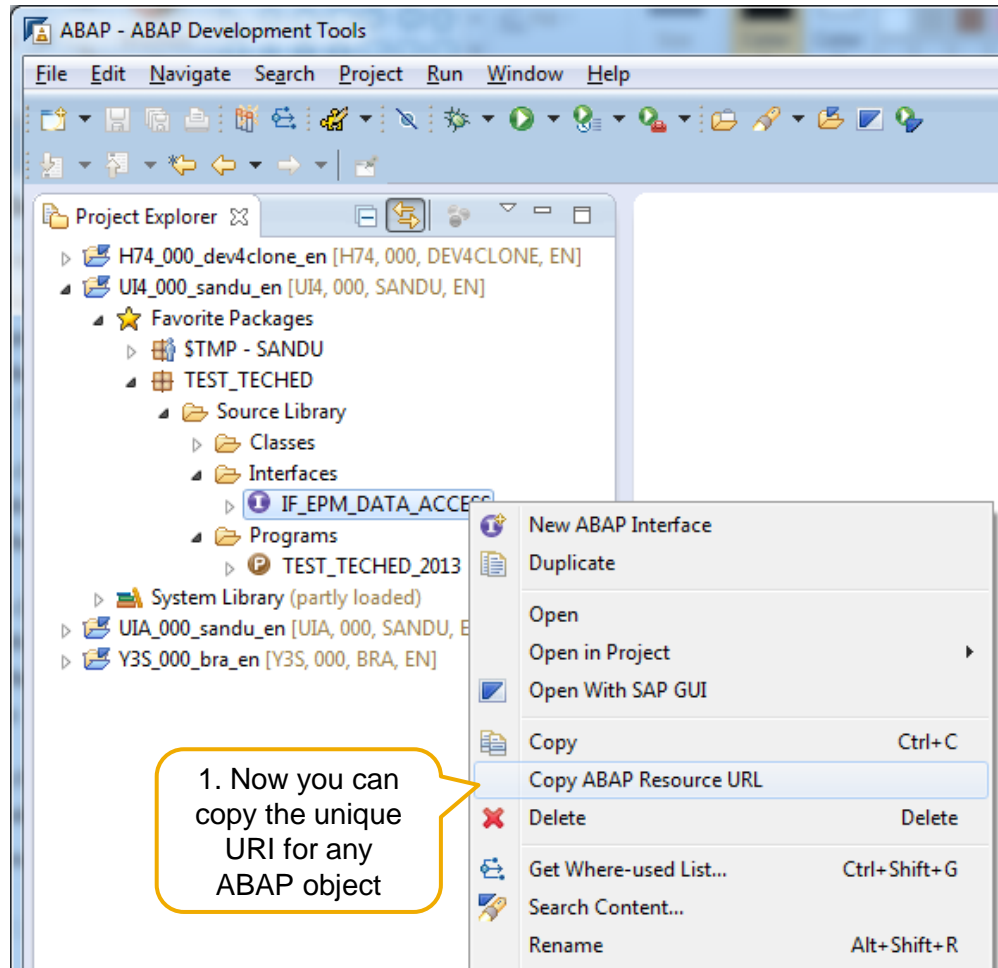
The image shows two screenshots from the ABAP Development Tools interface. The left screenshot shows the 'Window' menu open, with 'Preferences' selected at the bottom. A yellow callout box with the text '1. Go to preferences and ...' points to the 'Preferences' menu item. The right screenshot shows the 'Preferences' dialog box, specifically the 'ABAP Development' section. Under 'URL Handling', the checkbox 'Open ABAP resource URLs from external documents in this installation of ABAP Development Tools' is checked. A yellow callout box with the text '2. ...tick this option' points to this checkbox. The dialog also shows other settings like 'Handler', 'Show Sub Object Name in Editor Tab', and 'Input Field History Settings'.

1. Go to preferences and ...

2. ...tick this option

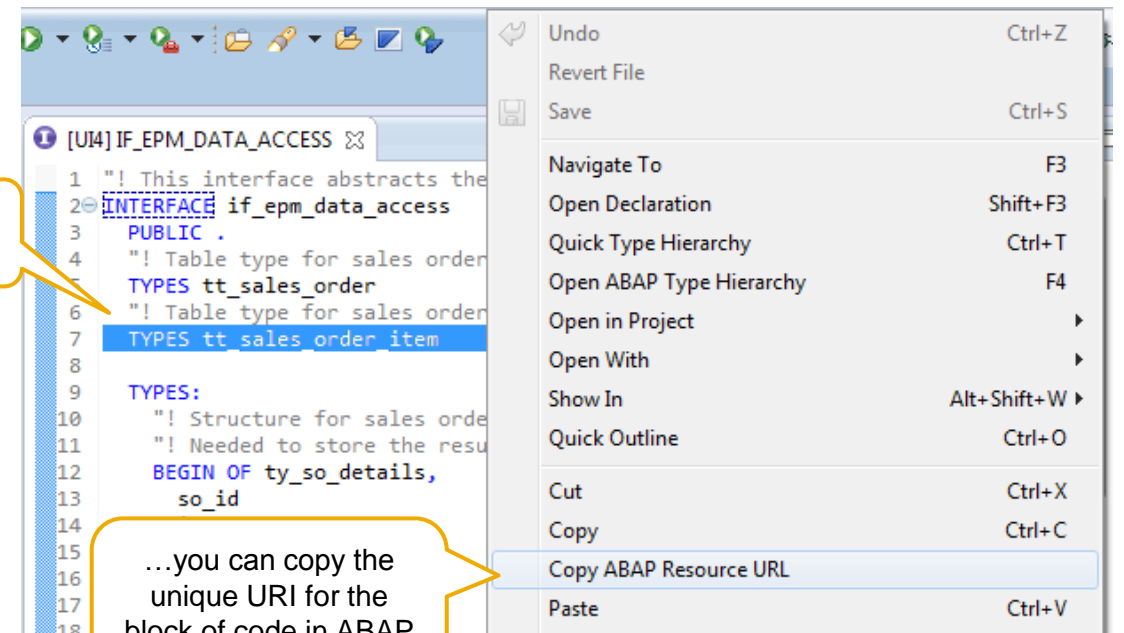
ABAP Resource URL of Development Objects

Copying ABAP Resource URL



1. Now you can copy the unique URI for any ABAP object

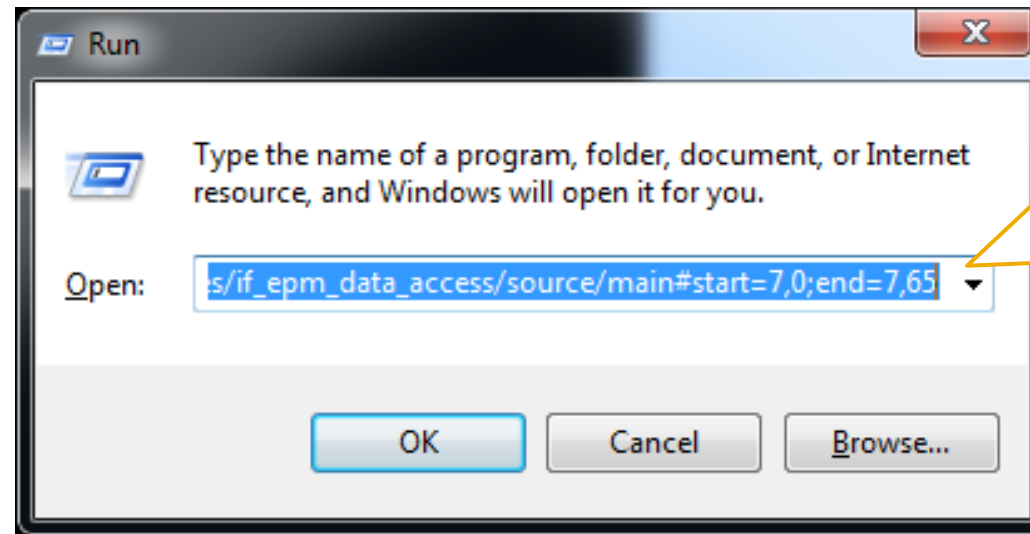
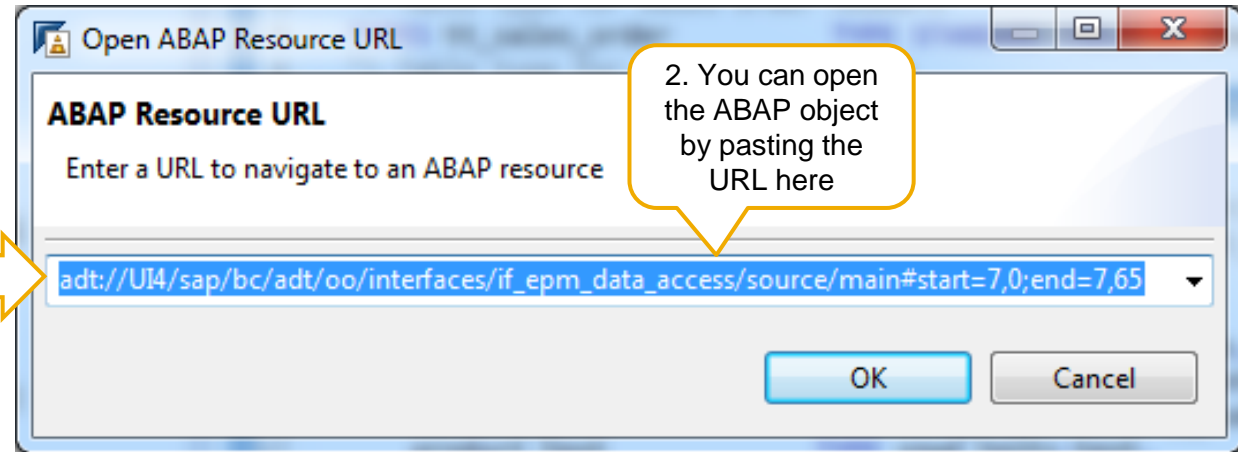
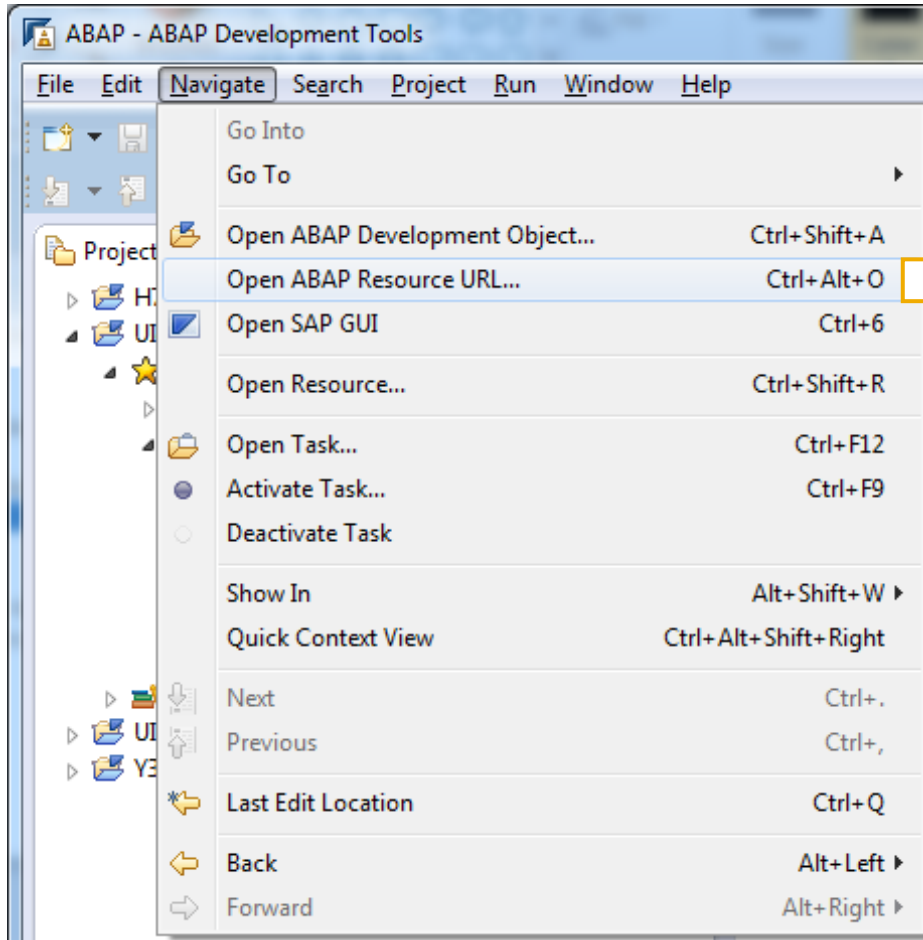
1. Select a block of lines and...



...you can copy the unique URI for the block of code in ABAP Class\Interface\Report

ABAP Resource URL of Development Objects

Using ABAP Resource URL



Searching Usages (Where-used)

1. To search for used references of any object, select the object and choose "Get Where-used List..."

2. All the ABAP objects referring to the "searched" object is listed in the search view.

3. Double click on the entry to open the respective editor and navigate to the correct line.

Line number and line details are also shown



Refactoring ABAP Source Code

Applying ABAP Quick Fixes

Refactoring ABAP Source Code

Ctrl+1
Quick Fix
Suggest possible fixes for a problem

1. Select any non ABAP keyword and press ctrl+1 to bring up the quick fix

For the ABAP variable in private section, quick fix offers these operations

- Rename sut (Alt+Shift+R)
- Rename sut in source unit (Ctrl+2, R)
- Make sut protected
- Make sut public
- Delete sut (Delete)

2. Select "Rename sut in source unit"... <next slide>

Renames **sut** and adjusts all occurrences of **sut** in the current source unit.
If **sut** is also used in other source units, the refactoring will be cancelled.

Press 'Tab' from proposal table or click for focus

Renaming ABAP Variables

Refactoring ABAP Source Code

```
*[UI4] TEST_TECHED_2013
92
93  ENDClass.
94
95  Start-of-Selection.
96
97  NEW lcl_main( )->run( ).
98
99  "! unit test for local lcl_main class using the stub implementation to make
100 CLASS ltcl_unit_test DEFINITION FINAL FOR TESTING
101   DURATION SHORT
102   RISK LEVEL HARMLESS.
103
104   PRIVATE SECTION.
105   DATA sut_new_name TYPE REF TO if_epm_data_access.
106   METHODS:
107     first_test FOR TESTING RAISING cx_static_check.
108   ENDClass.
109
110
111 CLASS ltcl_unit_test IMPLEMENTATION
112
113   METHOD first_test.
114
115     CREATE OBJECT sut_new_name TYPE cl_epm_data_access_stub.
116     NEW lcl_main( sut_new_name )->get_data( IMPORTING sales_orders = DATA(sales_orders)
117     cl_abap_unit_assert=>assert_equals( exp = 4 act = lines( sales_orders ) ).
118   ENDMETHOD.
119
120
```

3. ...and type a new name.

4. All its occurrences are automatically renamed

Renaming ABAP Methods

Refactoring ABAP Source Code

The screenshot shows the SAP IDE interface with an ABAP source code editor. The code is for a class named [UI4]IF_EPM_DATA_ACCESS. The method 'select_product_for_so_item' is selected. A yellow callout box with the text 'Ctrl+1 Quick Fix Suggest possible fixes for a problem' is positioned over the top right of the editor. A yellow callout box with the text '1. Select a method and press ctrl+1 to bring up quick fix' points to the selected method. A yellow callout box with the text '2. Choose Rename... <next slide>' points to the 'Rename select_product_for_so_item' option in the Quick Fix menu. The Quick Fix menu is open, showing three options: 'Rename select_product_for_so_item (Alt+Shift+R)', 'Rename select_product_for_so_item in source unit (Ctrl+2)', and 'Delete select_product_for_so_item (Delete)'. A yellow callout box with the text 'Starts the refactoring wizard to rename select_product_for_so_item.' is positioned to the right of the menu. The status bar at the bottom shows 'Smart Insert 44 : 20' and 'Press 'Tab' from proposal table or click for focus'.

```
28     max_rows                TYPE i DEFAULT 1000
29     EXPORTING
30     et_sales_orders         TYPE tt_sales_order
31     "! Reads sales order items for a given sales order
32     "! @parameter so_guid | The GUID of the sales order header
33     "! @parameter et_sales_order_items | Sales order items read from the persistence
34     METHODS
35     select_sales_order_items
36     IMPORTING
37     so_guid                  TYPE snwd_so-node_key
38     EXPORTING
39     et_sales_order_items    TYPE tt_sales_order_item.
40     "! Reads product information for a given sales order item
41     "! @parameter product_guid | The GUID for the sales order item
42     "! @parameter product | Product information read from the persistence
43     METHODS
44     select_product_for_so_item
45     IMPORTING
46     product_guid            TYPE snwd_so-node_key
47     EXPORTING
48     product                 TYPE snwd_so-node_key
49
50     "! Reads the text
51     "! @parameter text
52     "! @parameter lang
53     "! @parameter text
54     METHODS
55     select_text
56     IMPORTING
57     text_euid
```

Ctrl+1 Quick Fix
Suggest possible fixes for a problem

1. Select a method and press ctrl+1 to bring up quick fix

2. Choose Rename... <next slide>

Starts the refactoring wizard to rename select_product_for_so_item.

Smart Insert 44 : 20 Press 'Tab' from proposal table or click for focus

Renaming ABAP Methods

Refactoring ABAP Source Code

The image shows three sequential screenshots of the SAP 'Rename Method' dialog box, illustrating the steps to rename an ABAP method and its references.

Step 1: Choose Name
The 'New name' field contains 'select_product_for_sales_item'. A callout bubble says: "3. Enter new method name and press next".

Step 2: Choose Transport
The 'Affected objects' list includes:

- CL_EPM_DATA_ACCESS_STUB (Global Class)
- CL_EPM_DATA_ACCESS (Global Class)
- IF_EPM_DATA_ACCESS (Interface)
- TEST_TECHED_2013 (Program)

A callout bubble says: "Lists all the ABAP objects using the method". The 'Next >' button is highlighted, with a callout bubble saying: "4. Press Next".

Step 3: Rename Method
The dialog shows the changes to be performed in a table:

Original Source	Refactored source
28	28
29	29
30 product for so item.	30 product for sales item.
31 NTO product WHERE node_key =	31 NTO product WHERE node_k
32	32
33	33
34	34
35 _text.	35 _text.
36 text_key.	36 text_key.
37	37

A callout bubble says: "Shows the change in affected objects if method is renamed". The 'Finish' button is highlighted, with a callout bubble saying: "5. Upon Finish, the method is renamed and all the ABAP objects calling the method are updated automatically".



Comparing resources, Working with history

Working with Revision History

```
[UI4] TEST_TECHED_2013
1 REPORT test_teched_2013.
2
3 CLASS lcl_main DEFINITION CREATE
4
5
6 PUBLIC SECTION.
7
8 METHODS constructor
9 IMPORTING
10 p_data_access TYPE
11
12 METHODS run.
13 METHODS get_data EXPORTING
14
15 PROTECTED SECTION.
16 PRIVATE SECTION.
17 DATA data_access TYPE REF TO
18
19 ENDCLASS.
20
21 CLASS lcl_main IMPLEMENTATION.
22
23 METHOD constructor.
24 IF p_data_access IS NOT INITI
25 me->data_access = p_data_a
26 ELSE.
27 me->data_access = NEW cl_e
28 ENDIF.
29 ENDMETHOD.
30
31 METHOD run.
32 get_data(
33 IMPORTING
```

Cut Ctrl+X
Copy Ctrl+C
Copy ABAP Resource URL
Paste Ctrl+V
Get Where-used List... Ctrl+Shift+G
Search Content...
Rename Alt+Shift+R
Quick Fix Ctrl+1
Source Alt+Shift+S
Profile As
Debug As
Run As
Coverage As
Validate
Team
Compare With
Replace With
Unlock Ctrl+U
Preferences...
Remove from Context Ctrl+Alt+Shift+Down

For any ABAP Class\Interface\Report\Function Module, you can view the

- local history
- Revision History or
- Compare with version in another system

Local History
Each Other
Revision History...
H74_000_dev4clone_en [H74, 000, DEV4CLONE, EN]
UIA_000_sandu_en [UIA, 000, SANDU, EN]
Y3S_000_bra_en [Y3S, 000, BRA, EN]

1. Select revision history... <next slide>

Working with Revision History

Comparing ABAP resources with Compare Editor

The image shows a sequence of steps in the SAP IDE to compare two versions of an ABAP program:

1. The History view shows a list of revisions for the program [UI4] ZCD260_TEST_TECHED_FASTER (Program).

Revision	Date	Author	Transport Request	Description
Inactive	30.01.2014 13:30:33	SANDU		
Active	24.01.2014 14:31:06	SANDU		
2. The list of historical versions is displayed, showing the Inactive and Active versions.
3. Two versions are selected, and the "Compare" option is chosen from the context menu.
 - Copy (Ctrl+C)
 - Compare with Each Other
4. The "Compare" editor is opened, displaying the differences between the Inactive and Active versions side-by-side.

```
[UI4] ZCD260_TEST_TECHED_FASTER (Program) - Inactive
49  ls_pd TYPE snwd_pd.
50
51
52  me->data_access->select_sales_orders(
53  EXPORTING
54  max_rows = 100
55  IMPORTING
56  et_sales_orders = lt_sales_order ).
57
58  LOOP AT lt_sales_order INTO DATA(ls_sales_or
59
60  CLEAR ls_so.
61  ls_so-so_id = ls_sales_order-so_id.
62
63  me->data_access->select_sales_order_items(
64  EXPORTING
65  so_guid = ls_sales_order
66  IMPORTING
67  et_sales_order_items = DATA(lt_so_iter
68
69  LOOP AT lt_so_items INTO ls_so_item.
70
71  me->data_access->select_product_for_so:
72  EXPORTING
73  product_guid = ls_so_item-product_g
74  IMPORTING
75  product = ls_pd ).
76
77  me->data access->select text(

[UI4] ZCD260_TEST_TECHED_FASTER (Program) - Active
43  METHOD get_data.
44
45
46  me->data_access->SELECT_SALES_ORDER_DETAIL
47  EXPORTING
48  MAX_ROWS = 1000
49  IMPORTING
50  SO_DETAILS = sales_orders
51  ).
52
53  ENDMETHOD.
54
55  ENDClass.
56
57  START-OF-SELECTION.
58  l_main( )->run( ).
59
60  "! unit test for local lcl_main class usin
61  CLASS ltcl_unit_test DEFINITION FINAL FOR TE
62  DURATION SHORT
63  RISK LEVEL HARMLESS.
64
65  PRIVATE SECTION.
66  METHODS:
67  first_test FOR TESTING RAISING cx_stat
68  ENDClass.
69
70
71
```

Annotations in the image include:

- "2. Shows list of all historical versions of the ABAP object" (pointing to the History table)
- "3. Select 2 versions and choose 'Compare'" (pointing to the context menu)
- "4. 'Compare' editor is opened" (pointing to the Compare Editor window)
- "differences" (pointing to the side-by-side code comparison)
- "Inactive version" (pointing to the left pane)
- "Active version" (pointing to the right pane)

Working with Local History

Comparing ABAP resources with the local history

The screenshot displays the SAP IDE interface with the following components:

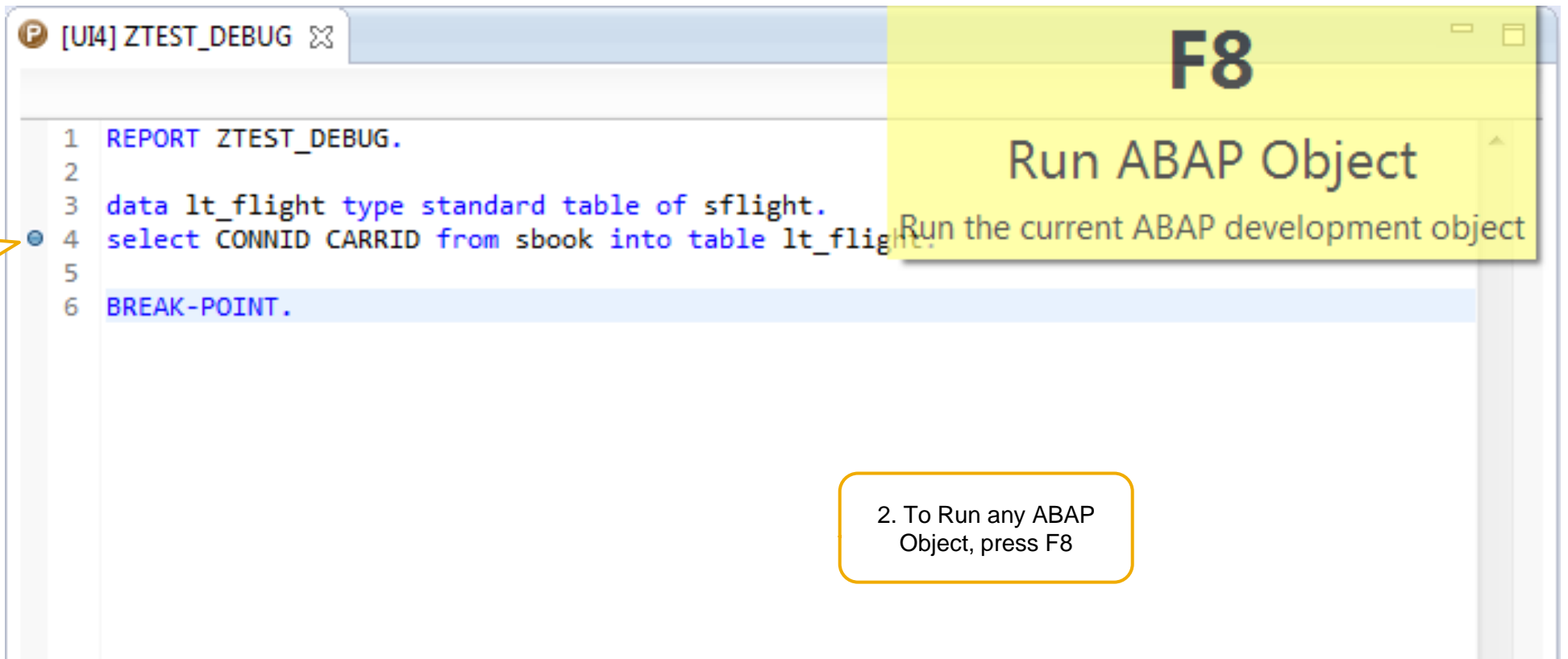
- Task List:** Shows the file `zcd260_test_teched_faster.asprog`.
- History Panel:** Lists revision times: 1/30/14, 1:30 PM; 1/24/14, 2:31 PM; 1/23/14, 7:26 PM; 1/23/14, 1:19 PM.
- ABAP Compare:** Compares two versions of the program `ZCD260_TEST_TECHED_FASTER`.
 - Local history version 1:** Shows code for `alv_table->display()` and `METHOD get_data` with a `DATA` section containing `lt_sales_order`, `ls_so`, `ls_so_item`, and `ls_pd`.
 - Local history version 2:** Shows the same code but with a `TODO: call the new performance optimize` comment added to line 46.
- Preferences - Local History:** Shows configuration options:
 - Limit history size
 - Days to keep files: 1
 - Maximum entries per file: 7
 - Maximum file size (MB): 1
 - Note: The 'Maximum entries per file' and the 'Days to keep files' values are only applied when compacting the local history on shutdown.

Annotations with arrows point to the History panel (2. Local history can also be viewed for any ABAP Class\Interface\Report etc.), the ABAP Compare window (Local history version 1 and 2), and the Preferences dialog (3. Configure local history settings).



Using Troubleshooting Tools

Debugging ABAP Code



The screenshot shows the SAP IDE interface for a report named ZTEST_DEBUG. The code is as follows:

```
1 REPORT ZTEST_DEBUG.  
2  
3 data lt_flight type standard table of sflight.  
4 select CONNID CARRID from sbook into table lt_flight.  
5  
6 BREAK-POINT.
```

A blue dot on line 4 indicates a breakpoint. A yellow tooltip is visible over the code, displaying the text "F8 Run ABAP Object Run the current ABAP development object".

1. Double click here and the breakpoint is set

2. To Run any ABAP Object, press F8

The Debug Perspective in Overview

Debugging ABAP Code

The screenshot displays the SAP ABAP Debug Perspective. On the left, a 'Confirm Perspective Switch' dialog box is open, asking if the user wants to open the Debug perspective. The main window shows the 'Debug - Program ZTEST_DEBUG [UI4] - active - UI4_000_sar' environment. The 'Source' view shows the following code:

```
1 REPORT ZTEST_DEBUG.  
2  
3 data lt_flight type standard table of sflight.  
4 select CONNID CARRID from sbook into table lt_flight.  
5  
6 BREAK-POINT.
```

The 'Variables' view on the right shows the following data:

Name	Value	Actual Type
<Enter variable>		
LT_FLIGHT	[1375240x14(112)] Standard Table	\TYPE=%_T00004S000000.
SY-SUBRC	0	SYST_SUBRC
Globals		

The 'Outline' view on the bottom right shows the 'ABAP Internal Table (Debugger)' for 'LT_FLIGHT - [1375240x14(112)] Standard Table'. The table contains the following data:

Row	MANDT	CARRID	CONNID
1	040	LH	0000
2	040	LH	0000
3	040	LH	0000
4	040	LH	0000
5	040	LH	0000
6	040	LH	0000
7	040	LH	0000
8	040	LH	0000
9	040	LH	0000

Annotations in the image include:

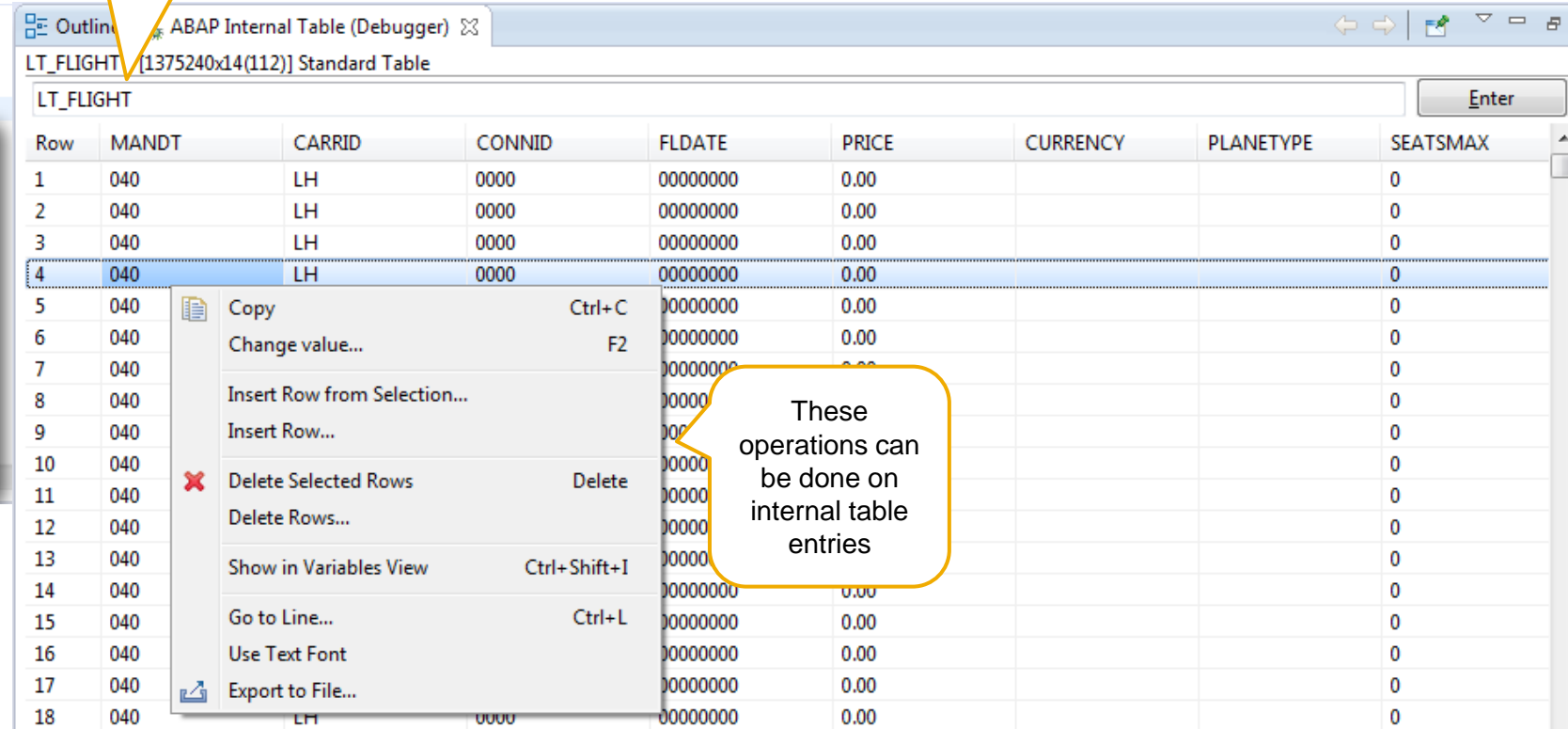
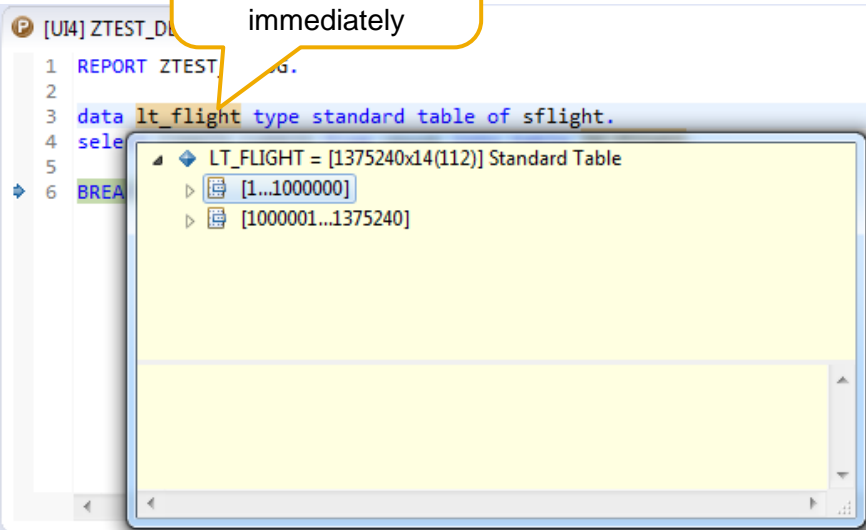
- A callout box with a question mark and text: "3. You will be asked to open Debug perspective. Choose Yes".
- A callout box with text: "4. Breakpoint is hit" pointing to line 6 of the code.
- Callout boxes for keyboard shortcuts: F5, F6, F7, F8, and "Go to line" pointing to the top toolbar.
- A callout box with text: "Variables view" pointing to the Variables view pane.

Displaying Internal Tables

Debugging ABAP Code

5. Place the cursor on a variable to view its value immediately

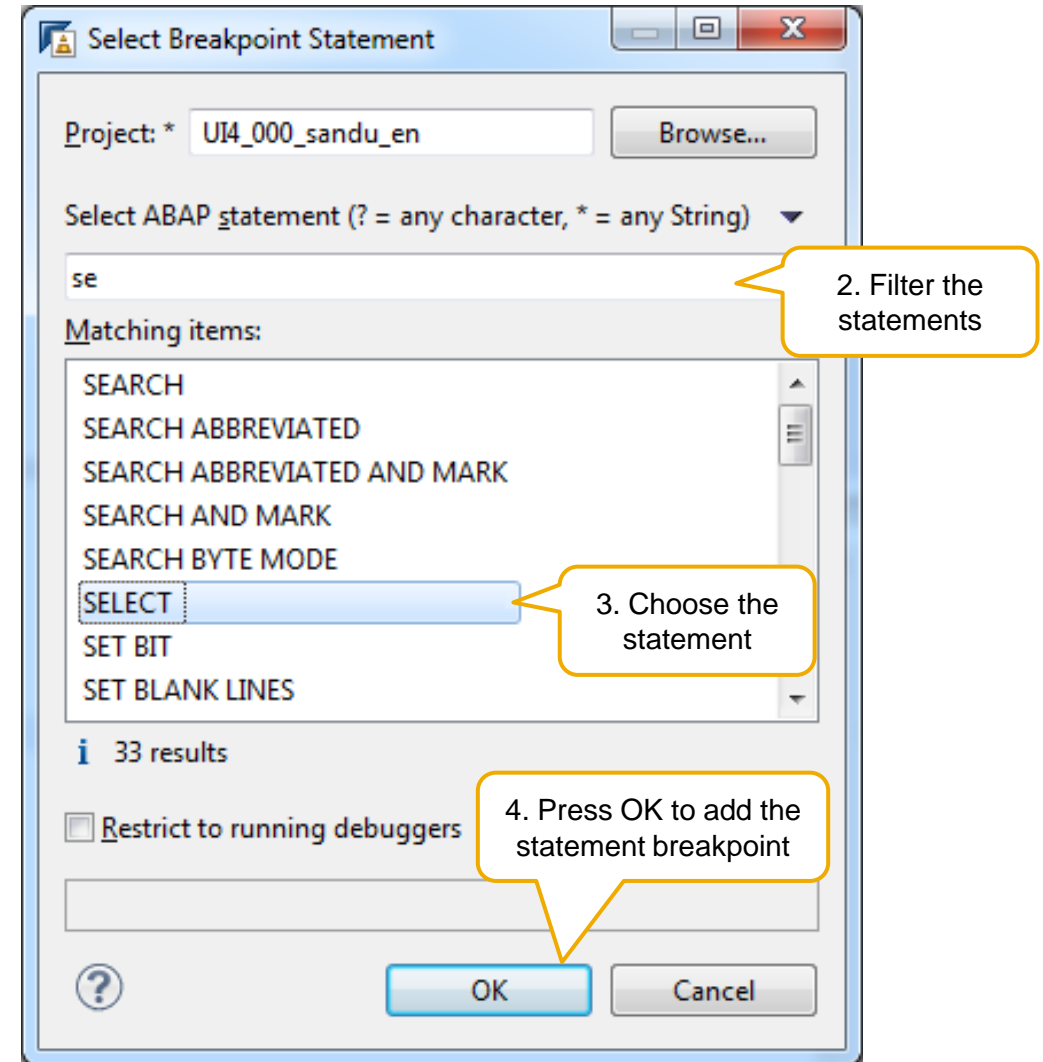
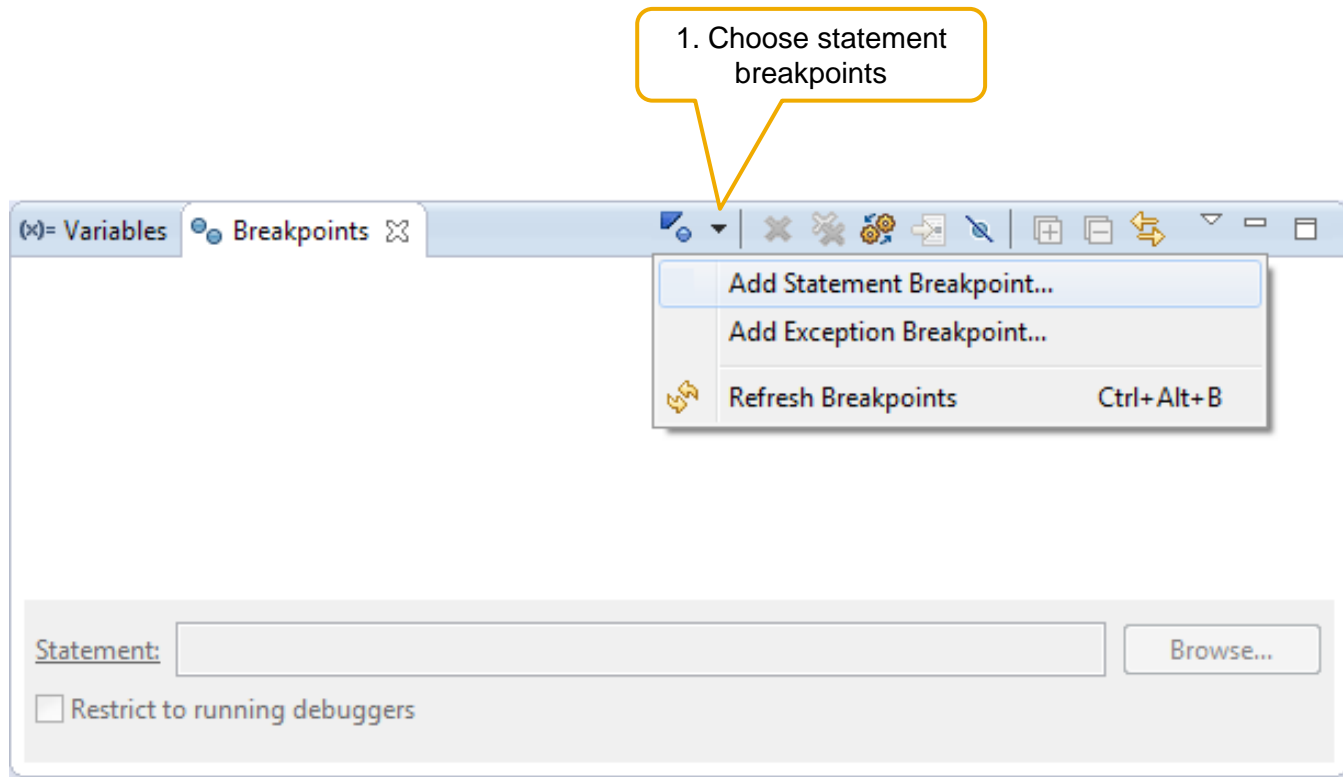
6. Type the table name and press enter to view its values



These operations can be done on internal table entries

Setting a Breakpoint for a statement

Debugging ABAP Code



Using ABAP Profiling – the ABAP trace

Tracing a Program

1. To profile any ABAP report, choose Profile As → ABAP Application

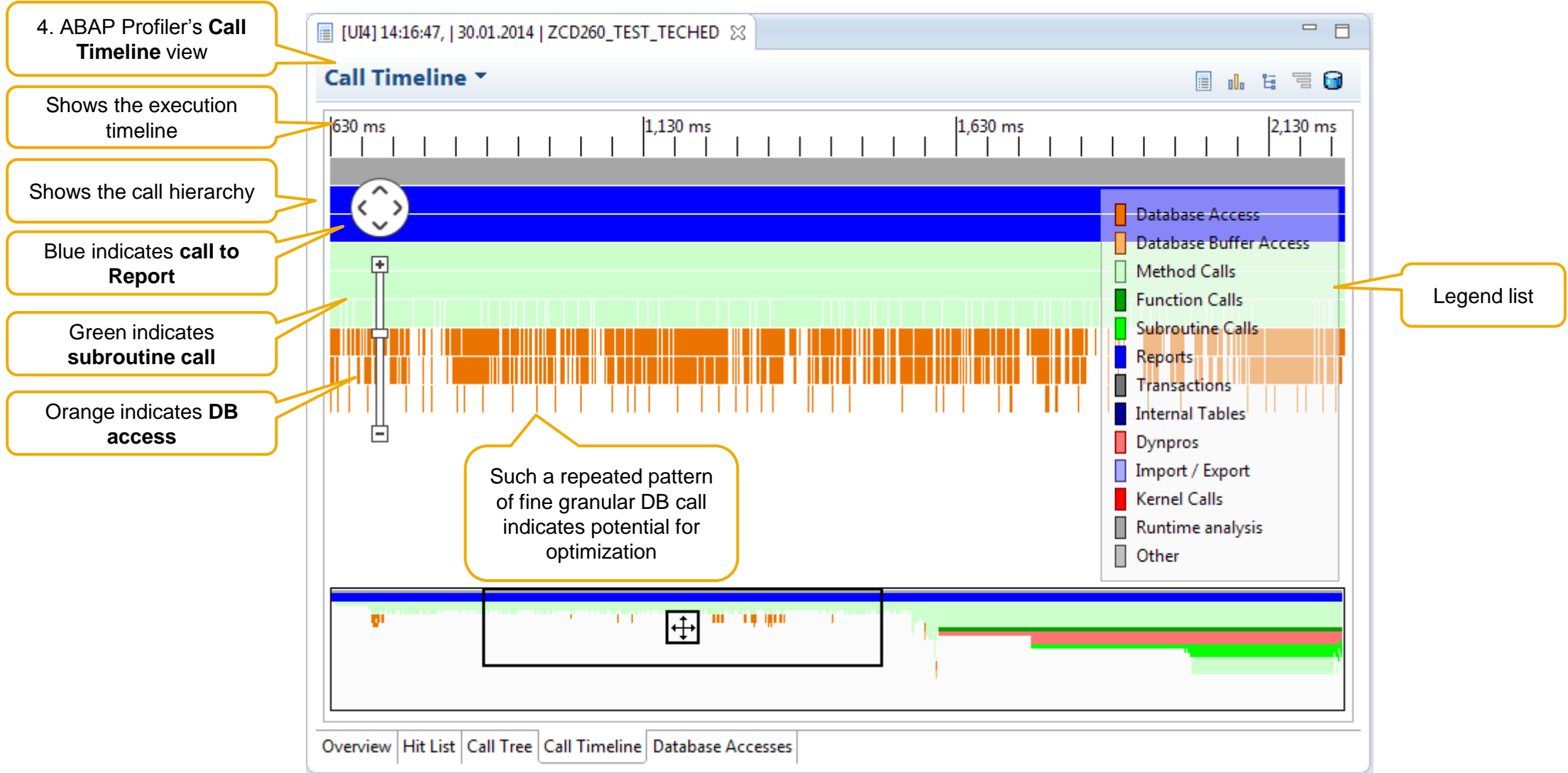
2. The trace can be accessed from the ABAP Traces view

3. Double-click... <next slide>

Project / Trace Title	Time	D
▶ H74_000_dev4clone_en [user filter: DEV4CLONE, refreshed: not yet]		
▶ UI4_000_sandu_en [user filter: SANDU, refreshed: 14:17:14, 30.01.2014]		
ZCD260_TEST_TECHED	14:16:47	30
ZCD260_TEST_TECHED_FASTER	14:31:16	24
ZCD260_TEST_TECHED	14:26:41	24
▶ UIA_000_sandu_en [user filter: SANDU, refreshed: not yet]		
▶ Y3S_000_bra_en [user filter: BRA, refreshed: not yet]		

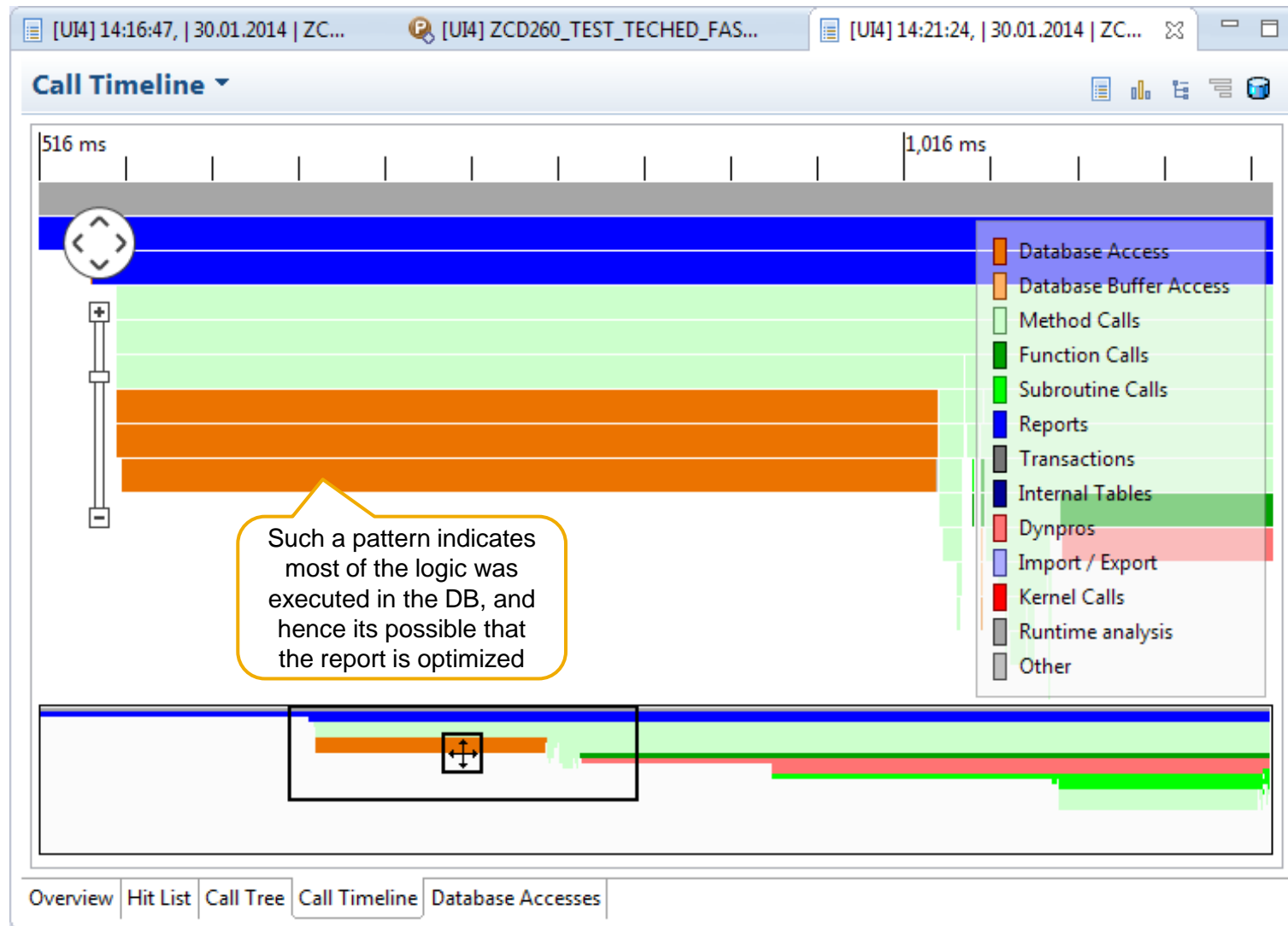
Understanding the Call Timeline

Using ABAP Profiling – the ABAP trace



Understanding the Call Timeline

Using ABAP Profiling – the ABAP trace





Transport Organizer

Using the Transport Organizer View

1. Lists all the system, its
 □ transport Requests
 □ transport Tasks and
 □ objects contained it

Transport Request	Owner	Type
▶ H74_000_dev4clone_e [user filter: DEV4CLONE, refreshed: not yet]		
▶ UI4_000_sandu_en [user filter: MORRISONI, refreshed: 14:09:56, 30.01.2014]		
▶ UIA_000_sandu_en [user filter: SANDU, refreshed: not yet]		
▶ Y3S_000_bra_en [user filter: MORRISONI, refreshed: 14:12:51, 30.01.2014]		
Workbench		
Modifiable		
Y3SK006202	OTTOF	
Y3SK052563	RITTERTH	
RIT_TEST_TRANS_RELEASE		Program
Y3SK006271	MORRISONI	
Y3SK006298	SCHULENBURG	
Y3SK017558	HUESTER	
Y3SK017645	HUESTER	
Y3SK017646	HUESTER	
Y3SK057715	MORRISONI	
Y3SK006215	CRASOVAN	
Y3SK006200	CLOCKNER	

2. These operations can be done for any Transport Request/Task

- Add User
- Copy (Ctrl+C)
- Delete
- Refresh (F5)
- Run Consistency Checks
- Change Owner
- Release

Transport Requests and tasks

Transport request/task owner and description



F1 Help

F1 Context Specific Help

Accessing ABAP in Eclipse Tools Help

The screenshot displays the Eclipse IDE interface. The main editor window is titled "[Y3S] COMPONENTCONTROLLER | WDC". The left sidebar shows the "Context" view with a "Context Structure" tree containing a single node "CONTEXT (1..1)". The right sidebar shows the "Help" view with a search bar and a list of related topics. A yellow callout bubble points to the "CONTEXT (1..1)" node in the context structure, containing the text: "Press F1 anywhere to bring up the context specific Help. In this case, all the help content for WDA context is displayed".

Context

Context Structure

type filter text

CONTEXT (1..1)

Used Controller

type filter text

Overview Context Attributes Methods Events Implementation

Help

Contents Search Related Topics

Bookmarks Index

About Controller Editor

Use this page to define the context hierarchy for the Component controller.

See also:

- Creating Controller Context
- Mapping Context
- Updating Context Mapping
- Using Dictionary Structure
- Context
- Context Properties
- Context Mapping
- Component Controller
- Working with Component Controller

More results:

- Search for Controller Editor

F1 ABAP Language Help

Accessing ABAP Help

The screenshot displays two windows from the SAP IDE. The left window, titled "[UI4] ZCD260_TEST_TECHED_FASTER", shows ABAP code for a class. The right window, titled "ABAP Language Help", displays the help content for the "METHODS" keyword. A callout bubble points to the "METHODS" keyword in the code.

```
1 REPORT zcd260_test_teched_faster.
2
3 CLASS lcl_main DEFINITION CREATE PUBLIC.
4
5
6 PUBLIC SECTION.
7
8 METHODS constructor
9     IMPORTING
10         p_data_access TYPE REF TO zif_epm_data.
11 METHODS run.
12 METHODS get_data EXPORTING sales_orders TYPE REF TO zif_epm_data.
13 PROTECTED SECTION.
14 PRIVATE SECTION.
15 DATA data_access TYPE REF TO zif_epm_data.
16
17 ENDCLASS.
18
19 CLASS lcl_main IMPLEMENTATION.
20
21 METHOD constructor.
22     IF p_data_access IS NOT NULL.
23         me->data_access = p_data_access.
24     ELSE.
25         me->data_access = NEW zcl_epm_data_acc.
26     ENDIF.
27 ENDMETHOD.
28
29
30 METHOD run.
31     get_data(
32         IMPORTING
```

ABAP Language Help

Entry page > ABAP Help > ABAP - Reference > Declarations > Declaration Statements > Classes and Interfaces > Components in Classes and Interfaces > Methods >

METHODS

[Short Reference](#)

Syntax Forms

[General Instance Methods](#)

```
1.METHODS meth [ABSTRACT|FINAL]
   [IMPORTING parameters [PREFERRED
PARAMETER p]]
   [EXPORTING parameters]
   [CHANGING parameters]
   [{RAISING exc1|RESUMABLE(exc1)
exc2|RESUMABLE(exc2) ...}
   |{EXCEPTIONS exc1 exc2 ...}].
```

[Functional Instance Methods](#)

```
2.METHODS meth [ABSTRACT|FINAL]
   [IMPORTING parameters [PREFERRED
PARAMETER p]]
   [EXPORTING parameters]
```

Place cursor on ABAP keyword and press F1 to bring up the ABAP Language Help



Web Dynpro ABAP Development Tools

View Editor

The Real WYSIWYG Editor

The screenshot displays the SAP View Editor interface, split into two panes. The left pane shows the design-time view, and the right pane shows the runtime view of the same application.

Design Time WDA View (Left Pane):

- Layout:** Shows a table with 4 columns: Airline, Airline, Airline Currency, and URL. The table is currently empty.
- Favorite Book:** A form with fields for Category (dropdown), Title, Number of pages, ISBN, and Author.
- Display Changes:** A table with 8 columns: Name, Number, Path, Change Type, Index, Attribute, Old Value, and New Value. The table contains one row with '0' in the Number and Index columns.
- Buttons:** Enter, Display and Cancel Changes, Activate changes, Deactivate Changes.
- Footer:** Overview | Layout | Context | Attributes | Methods | Plugs | Implementation

Runtime View of WDA Application (Right Pane):

- URL:** https://ldai2uia.wdf.sap.corp:44300/sap/bc/webdynpro/sap/demo_coi
- Layout:** Shows the same table and form as the design view, but with data populated.
- Airlines Table:**

Airline	Airline	Airline Currency	URL
AA	American Airlines	USD	http://www.aa.com
AC	Air Canada	CAD	http://www.aircanada.
AF	Air France	EUR	http://www.airfrance.fi
AZ	Alitalia	EUR	http://www.alitalia.it
BA	British Airways	GBP	http://www.british-airv
- Favorite Book Form:** Category: Book, Title: Very new Book, Number of pages: 111, ISBN: ISBN_4711, Author: Norman Unknown.
- Display Changes Table:** Empty table.
- Buttons:** Enter, Display and Cancel Changes, Activate changes, Deactivate Changes.

Design time WDA view in View Editor

Runtime view of WDA Application

This is the real WYSIWYG editor!!

Context Editor

Visual Context Mapping

The screenshot displays the SAP Context Editor interface. On the left, the 'Context Structure' pane shows a hierarchy starting with 'CONTEXT (1..1)' and a sub-context 'NODE_CONN_POPUP (1..1)'. This sub-context has five attributes: 'ATT_ACCOUNT : STRING', 'ATT_ACCOUNT_TYPE : SRT_WSP_ACCOUNT_TYPE', 'ATT_POPUP : IF_WD_WINDOW', 'ATT_PROVIDER_IBCR : SRT_WSP_IBCR_ID', and 'ATT_SYSTEM_GUID : GUID_32'. Below this is another sub-context 'NODE_RESULTS (0..n)' with attributes 'ATT_CLOSE_ENABLED : WDY_BOOLEAN', 'ATT_PROGRESS : INT4', and 'ATT_TIMEOUT : INT4'. On the right, the 'Component Controller' pane shows a similar hierarchy for 'C_SRT_ACCOUNT_MANAG.COMPONENTCONTROLLER', with a sub-context 'NODE_CONN_POPUP (1..1)' that has three attributes: 'ATT_ACCOUNT : STRING', 'ATT_ACCOUNT_TYPE : SRT_WSP_ACCOUNT_TYPE', and 'ATT_POPUP : IF_WD_WINDOW'. Yellow arrows indicate the mapping of these attributes from the left context to the right one. A callout box labeled 'Context Hierarchy' points to the left pane, 'Visual Context Mapping' points to the arrows, and 'Drag and drop to map context' points to the right pane. At the bottom, the 'Properties' pane shows details for 'NODE_CONN_POPUP'.

Property	Value
Name	NODE_CONN_POPUP
Dictionary Structure	
Cardinality	1..1
Selection	0..1
Initialization of Lead Selection	True
Singleton	False
Supply Function	
Mapping Path	C_SRT_ACCOUNT_MANAG.COMPONENTCONTROLLER.NODE_CON...

Context Editor

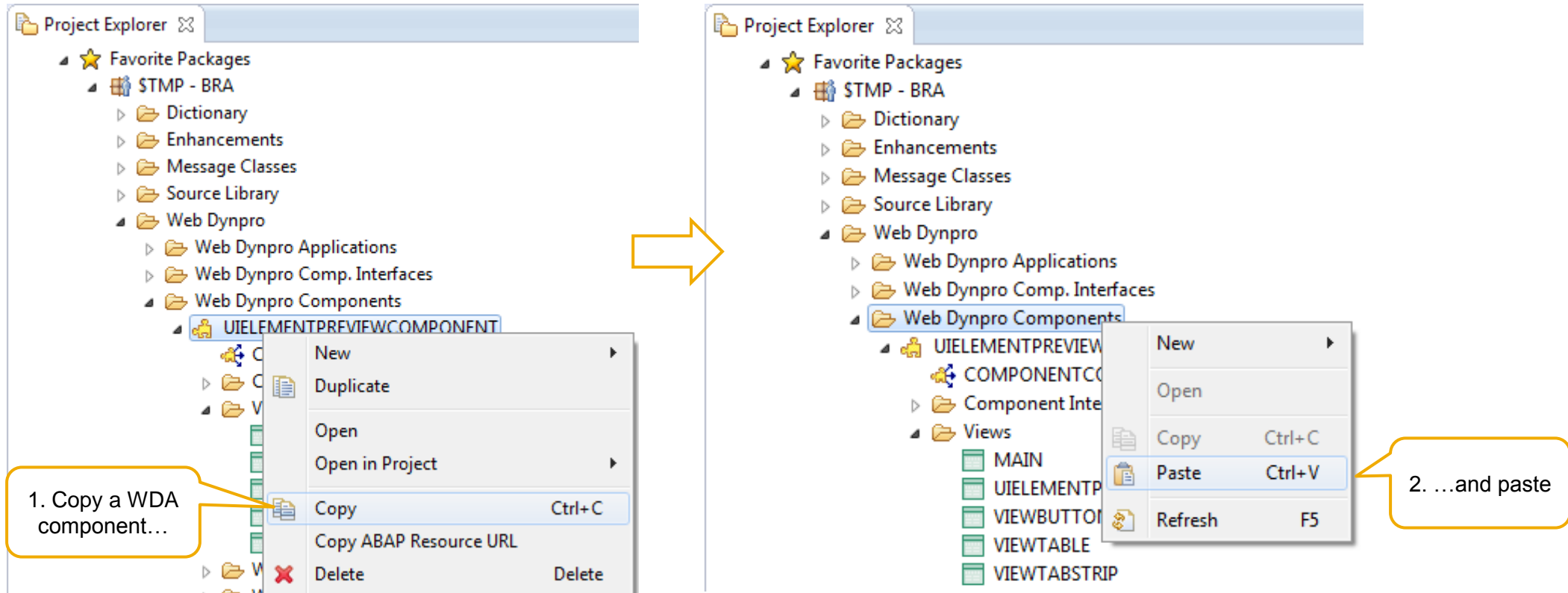
Visual Context Binding

The image illustrates the process of visual context binding in the Context Editor. It is divided into three main sections:

- 1. Select any UI element and choose Bind to Context:** The 'Outline' view on the left shows a tree structure of UI elements. The 'ROOTUIELEMENTCONTAINER' is selected, and a context menu is open with the 'Bind to Context...' option highlighted. A callout box points to this menu item.
- 2. Visual Context binding:** The 'Bind to Context' dialog is shown in the center. It has two panes: 'Layout' on the left and 'Context' on the right. The 'Layout' pane lists various UI properties like 'AccessibilityDescription', 'AriaLandmark', 'DefaultButtonId', etc. The 'Context' pane shows a tree of context nodes and attributes, with 'ATT_TIMEOUT : INT4' selected. Dashed lines connect the selected property in the 'Layout' pane to the selected attribute in the 'Context' pane. A callout box points to these connections.
- 3. Drag and drop to bind context:** A large curved arrow indicates the final step of the process, from the 'Layout' pane to the 'Context' pane. A callout box points to this arrow.

At the bottom of the dialog, there are 'Finish' and 'Cancel' buttons.

Copy & Paste of Component, Component Interface....



Copy & Paste of Context

The image displays two instances of the SAP IDE's 'Context' editor. The top instance shows the 'Context Structure' pane with a tree view containing nodes like 'CONTEXT (1..1)', 'NODE_ADMIN_DATA (1..1)', 'NODE_ASSIGNS (0..n)', 'NODE_BUTTONS_ENBL (1..1)', 'NODE_CONN_POPUP (1..1)', 'NODE_FILTER_ASSIGNS (1..1)', 'NODE_MODE (1..1)', 'NODE_UI_CTRL (1..1)', and 'UI (1..1)'. A context menu is open over the 'CONTEXT (1..1)' node, with the 'Copy' option selected. A callout bubble points to this menu with the text '1. Copy context...'. The bottom instance shows the same 'Context Structure' pane, but with a context menu open over the 'CONTEXT (1..1)' node, with the 'Paste' option selected. A callout bubble points to this menu with the text '2. ...and paste in any controller editor, in any Component, across ABAP systems'. The 'Used Controller' pane in the bottom instance shows the copied context structure being pasted into the controller's context.

1. Copy context...

2. ...and paste in any controller editor, in any Component, across ABAP systems

Copy & Paste of Methods, Attribute, events

The image shows two side-by-side SAP IDE windows. The left window displays the 'Methods' overview for a controller, with a context menu open over the 'WDDOMODIFYVIEW' method. A yellow callout box points to the 'Copy' option, labeled '1. Copy Method...'. The right window shows the same controller with the 'WDDOINIT' method selected. A yellow callout box points to the 'Paste' option in its context menu, labeled '2. ...and paste in any controller editor, in any Component, across ABAP systems. Method definition and implementation is consistently copied'. A yellow arrow points from the 'Copy' callout in the left window to the 'Paste' callout in the right window. The 'Method Details' panel on the right shows the name 'WDDOINIT' and description 'Controller Initialization Me'.

ADT Shortcut Reference Card

Edit

Ctrl+Shift+A	Open development object
Ctrl+F2	Check development object
Ctrl+F3	Activate development object
Ctrl+Shift+F3	Activate all inactive objects
Ctrl+Space	Code completion
Ctrl+1	Quick fix proposal
Alt+F2	Show method information
Ctrl+7	Toggle comment
Shift+F1	Format source aka pretty printer

Help

F1	ABAP keyword documentation
F2	Show code element information
Ctrl+3	Search for commands & views
Ctrl+Shift+L	List all keyboard shortcuts

Navigate

F3	Open definition
Alt+Left	Backward history
Alt+Right	Forward history
Ctrl+T	Quick hierarchy
F4	Open Type Hierarchy
Ctrl+O	Quick outline
Ctrl+Shift+G	Where-used list

Run, Debug

F8	Run current ABAP object
Alt+F8	Select & run ABAP application
Ctrl+Shift+B	Toggle breakpoint
F5, F6, F7, F8	Step into, over, return, resume
Ctrl+Shift+F10	Execute ABAP unit tests
Alt+F9	Profile development object

Further Information

SAP Public Web

<http://scn.sap.com/community/abap/eclipse>

ADT 2.7

<http://scn.sap.com/docs/DOC-41168>

ADT 2.13

<http://scn.sap.com/community/abap/eclipse/blog/2013/07/16/new-version-of-abap-development-tools-for-eclipse-213-available-now>

ADT 2.16

<http://scn.sap.com/community/abap/eclipse/blog/2013/10/01/new-adt-version-216-for-eclipse-kepler-now-available-in-sap-service-marketplace>

ADT 2.19

<http://scn.sap.com/community/abap/eclipse/blog/2013/12/20/get-abap-development-tools-219-and-find-out-whats-new>

ADT 2.24

<http://scn.sap.com/community/abap/eclipse/blog/2014/03/28/abap-development-tools-version-224-available-now>



Thank You!

Contact Information:

Raghuvira Bhagavan

Product Expert, ABAP Development Tools

raghuvira.bhagavan@sap.com