Information Management software

IBM

DB2 for z/OS Administration Tools for Enhanced Change Management

- DBA activities with IBM DB2 Object Comparison and Administration Tool
- Change management functions with Version 7.2 of the two tools
- Scenarios and examples

Paolo Bruni
Lucy Krantz
Frank Nikodem
Guenter Pfann
Marc White

Archived

Redbooks

ibm.com/redbooks

Archived

International Technical Support Organization

DB2 for z/OS Administration Tools for Enhanced Change Management

June 2007

Archived

SG24-7441-00

Note: Before using this information and the product it supports, read the information in “Notices” on page xxiii.

Archived

First Edition (June 2007)

This edition applies to Version 7.2 of IBM DB2 Data Administration Tool for z/OS (program number 5697-L90) and IBM DB2 Object Comparison Tool for z/OS (program number 5697-L40).

© Copyright International Business Machines Corporation 2007. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures ix

Tables xix

Examples xxi

Notices xxiii
Trademarks xxiv

Preface xxv
The team that wrote this IBM Redbooks publication xxv
Become a published author xxviii
Comments welcome. xxviii

Part 1. Introduction 1

Chapter 1. Tools overview 3
1.1 Products: Overview 4
1.1.1 DB2 Administration Tool for z/OS 4
1.1.2 DB2 Object Comparison Tool for z/OS 6
1.2 Change Management support. 8
1.2.1 Change Management in general. 8
1.2.2 Change Management using DB2 Admin and DB2 Compare 9
1.2.3 Definition of new Change Management functionality 9
1.2.4 Installation considerations and prerequisites for using CM. 10
1.2.5 CM mandatory, optional, or not activated 11
1.2.6 Core functions for managing DB2 object changes 11

Part 2. Getting started with Change Management 13

Chapter 2. Customization 15
2.1 Installing DB2 Admin and DB2 Compare 16
2.1.1 z/OS related customization 16
2.1.2 Sizing the objects for Change Management 17
2.1.3 Creator for Change Management tables. 18
2.1.4 DB2 BINDs 18
2.1.5 Running the customization EXEC. 19
2.1.6 Invoking DB2 Admin 20
2.1.7 Enabling Change Management in your ISPF table. 23
2.1.8 Disabling Change Management 27
2.1.9 Specifying DB2 Compare on the DB2 Admin Main Menu. 27
2.1.10 Additional information for your ISPF table. 29
2.1.11 Running two different versions in parallel (co-existence) 29
2.1.12 Quick start customization summary 29
2.2 Maintenance strategy and deployment 31
2.2.1 Building a new SMP/E package 31
2.2.2 Restoring the flat file 32
2.2.3 Doing the BINDs 33
2.2.4 Invoking DB2 Admin with the new maintenance level. 33
2.2.5 Deploying your new maintenance level to another LPAR. 35

© Copyright IBM Corp. 2007. All rights reserved.

2.2.6 Advantages of this maintenance strategy . . . 35
2.3 Tailoring for Change Management . . . 35
2.3.1 Managing the SQL ID table . . . 35
2.3.2 Setting the default CM level . . . 38
2.3.3 Working with multiple Change Management Databases . . . 38
2.4 DB2 Security for Change Management . . . 39
2.4.1 Access to Change Management based on your ADBL Clist . . . 39
2.4.2 Access to Change Management based on DB2 security . . . 41
2.4.3 Additional security aspects . . . 42
2.4.4 Update of changes . . . 42
2.4.5 Security based on collection IDs . . . 43
2.4.6 Removing GRANTs . . . 43

Chapter 3. First steps with DB2 Admin . . . 45
3.1 First time in DB2 Admin . . . 46
3.1.1 Navigating the menus . . . 46
3.1.2 Customizing your DB2 Admin parameters . . . 48
3.2 Looking at database objects . . . 52
3.2.1 Navigating the DB2 system catalog . . . 54
3.2.2 Optimizing the SQL search . . . 64
3.2.3 Printing output to the print data set . . . 69
3.3 Creating new objects based on existing objects . . . 72
3.3.1 Creating a limited number of objects (DDL) . . . 73
3.3.2 Creating many objects including dependencies (GEN) . . . 76
3.4 Working with a copy of the DB2 system catalog . . . 79
3.5 Showing database structure and data set information . . . 81
3.6 Revoking impact . . . 83
3.7 Dropping impact . . . 85
3.8 Restarting the monitor . . . 88
3.9 Using work statement lists . . . 91
3.9.1 Work statement list basics . . . 91
3.9.2 Creating a work statement list . . . 92
3.9.3 Working with a work statement list . . . 93
3.9.4 Modifying a work statement list . . . 96
3.9.5 Examining a work statement list . . . 99
3.9.6 Executing a work statement list . . . 104
3.9.7 Cloning a work statement list . . . 105

Chapter 4. Working with the Change Management functions . . . 111
4.1 General considerations . . . 112
4.1.1 Point-and-shoot . . . 112
4.1.2 Data sets . . . 112
4.1.3 Version files . . . 112
4.2 Overview of Change Management . . . 113
4.2.1 Change Management (CM) offers a structured way to work with changes . . . 116
4.2.2 A change consists of a set of SQL statements . . . 118
4.2.3 Generating changes . . . 119
4.2.4 Set of changes can be promoted to other systems and imported there . . . 120
4.2.5 Changes are stored in a CM data base (one per subsystem) . . . 122
4.2.6 Answering questions such as “Who did what, when?” . . . 124
4.2.7 Immediate deployment possible for small changes . . . 126
4.2.8 Changes can be recovered . . . 126

4.3 Change Navigation . . . 128
4.3.1 Looking up a change . . . 129
4.3.2 Available commands from the change summary panel . . . 130
4.3.3 Showing change statements . . . 131
4.3.4 Refreshing your changes . . . 132
4.3.5 REPORT changes . . . 132
4.4 Change processes . . . 133
4.4.1 Status of a change . . . 134
4.4.2 The REGISTER process . . . 134
4.4.3 The ANALYZE process . . . 139
4.4.4 The RUN process . . . 145
4.4.5 The PROMOTE process . . . 151
4.4.6 The IMPORT process . . . 155
4.4.7 The RECOVER process . . . 158
4.5 Change status . . . 162
4.6 Change types . . . 163
4.7 Canceling a change . . . 164
4.8 Restarting . . . 164
4.9 Using the masking feature . . . 166
4.9.1 Field descriptions . . . 166
4.9.2 Migrating an existing masking data set . . . 167
4.9.3 Updating a masking entry in your Change Management database . . . 168
4.9.4 Masking during Compare . . . 168
4.9.5 SAVE handling for a mask . . . 168
4.10 Using the ignore feature . . . 169
4.10.1 Updating the ignored fields . . . 169
4.10.2 SAVE handling for an ignore . . . 171
4.11 Change terminology . . . 172
4.11.1 Generating a first change . . . 172
4.11.2 Pending change . . . 176
4.11.3 Further actions against a pending change . . . 180
4.11.4 Virtual change . . . 180
4.11.5 Emergency change . . . 180
4.11.6 Immediate change . . . 184
4.11.7 Prerequisite change . . . 185
4.11.8 Delta change . . . 186
4.12 Versions . . . 186
4.12.1 Base . . . 187
4.12.2 Delta version . . . 194
4.12.3 Versions: The whole picture . . . 196
4.13 Change flow with Change Management . . . 196
4.13.1 Change on the source environment using COMPARE . . . 196
4.13.2 Changes on the source environment using the ALTER ALC function . . . 197

Part 3. Change Management scenario . . . 199

Chapter 5. Implementing an initial design (V0) . . . 201
5.1 Sample IT database environment . . . 202
5.1.1 Development test . . . 203
5.1.2 Acceptance test . . . 203
5.1.3 Stress test . . . 203
5.1.4 Production . . . 204

Archived

5.2 Developing DDL using DB2 Admin . . . 204
5.2.1 Creating single objects (DDL) . . . 204
5.2.2 Creating many objects based on existing objects (GEN) . . . 204
5.3 Developing databases using generated DDL . . . 211
5.3.1 Methods of generating DDL . . . 211
5.3.2 Creating the Master Development Database (D0) . . . 211
5.4 Rolling out the initial design (V0) . . . 212
5.4.1 Using generate (GEN) to clone a database . . . 213
5.4.2 Creating (cloning) development databases (D1-D4) using MIG . . . 213
5.4.3 Cloning to another DB2 subsystem (MIGRATE) . . . 221
5.4.4 Cloning multiple physical databases (WSL cloning and COMPARE) . . . 227

Chapter 6. Implementing changes to a design (Vn) . . . 257
6.1 Modifying DB2 objects using DB2 Admin (ALC) and (AL) . . . 258
6.1.1 Creating a base version file (required) . . . 258
6.1.2 Classifying changes performed using DB2 Admin . . . 260
6.1.3 Altering database objects using the AL command . . . 264
6.1.4 Using the ALC command to implement table changes . . . 273
6.1.5 Registering a change through ALC . . . 282
6.1.6 ANALYZE: Analyzing a change . . . 283
6.1.7 RUN: Running a change and creating a new version file . . . 286
6.1.8 Changes in the technical design . . . 288
6.2 Modifying the Master Development Database (D0) using generated DDL and Object Compare . . . 292
6.2.1 Making changes to the database . . . 292
6.2.2 Generating and modifying DDL . . . 293
6.2.3 Using DB2 Object Compare panels . . . 294
6.2.4 Running a Compare job and using the CM function . . . 299
6.2.5 Registering a CHANGE . . . 299
6.2.6 ANALYZE and RUN . . . 301
6.2.7 Specify Ignore Fields (optional) . . . 301
6.2.8 Specify Mask Fields (optional) . . . 306

Chapter 7. Rolling out new versions of a database design . . . 311
7.1 Rolling out using the PROMOTE process . . . 312
7.1.1 The change being made . . . 312
7.1.2 Comparing the base versions . . . 313
7.1.3 Registering the change . . . 315
7.2 Rolling out using Compare . . . 317
7.2.1 The change being made . . . 318
7.2.2 Invoking DB2 Compare . . . 318
7.2.3 Specifying the compare source . . . 319
7.2.4 Specifying the compare target . . . 320
7.2.5 Specifying compare masks . . . 321
7.2.6 Specifying fields to ignore . . . 322
7.2.7 Generating a compare job . . . 322
7.3 Importing delta changes data set . . . 327
7.4 ANALYZE and RUN . . . 328

Chapter 8. Recovery of changes . . . 329
8.1 Recovery process in detail . . . 330
8.2 Recovery considerations for a completed change . . . 330
8.2.1 Recovering a completed change with existing data . . . 332
8.2.2 Recovering a completed change with original data . . . 346

8.3 Recovering an uncompleted change 354
8.4 Backing out to a previous version (without RECOVER process) 359
8.5 Drop Recovery 362

Chapter 9. Additional considerations 365
9.1 Performance considerations (scalability) 366
9.1.1 Large CM database 366
9.1.2 Additional indexes 366
9.1.3 Virtual changes 367
9.1.4 Using Data Archive Expert for z/OS 367
9.2 Diagnosis and troubleshooting 368
9.2.1 TSO ISPDPTRC 369
9.2.2 ADBDIAG DD statement 369
9.3 Data conversion 373
9.3.1 Stop conversion 375
9.3.2 Skip conversion 377
9.4 Limitations with CM 382

Related publications 383
IBM Redbooks 383
Other publications 383
Online resources 383
How to get IBM Redbooks 383
Help from IBM 384

Index 385

Archived

Archived

Figures

1-1 DB2 Admin Tool overview . . . 4
1-2 DB2 Compare overview . . . 7
1-3 Change Management process . . . 9
1-4 Core Change Management functions . . . 11
2-1 The definition of ADB2ATH and ADB2UTIL in your SYS1.PARMLIB . . . 16
2-2 Adapt the owner of your Change Management tables . . . 18
2-3 ADB2CUST Main Menu . . . 19
2-4 ADB2CUST General parameters . . . 20
2-5 A sample of the ADBL Clist . . . 21
2-6 A sample REXX™ to invoke ADBL including an argument . . . 22
2-7 DB2 Admin Main Menu . . . 23
2-8 ADB2CUST Main Menu . . . 23
2-9 Choosing the ISPF table . . . 24
2-10 Tag summary of your ISPF table . . . 24
2-11 The tag :cmown has been set to the value ‘A72’ . . . 25
2-12 Change Management tables . . . 26
2-13 DB2 Admin Main Menu with Change Management enabled . . . 26
2-14 Disabling Change Management . . . 27
2-15 Adding DB2 Compare to your ISPF table . . . 28
2-16 DB2 Admin Main Menu . . . 28
2-17 Maintenance strategy - Initial approach . . . 32
2-18 BIND PACKAGE and BIND PLAN . . . 33
2-19 Restoring the files and creating aliases . . . 34
2-20 The SQL ID table . . . 37
2-21 Inserting a new SQL ID . . . 37
2-22 SQL ID table - Default set to NONE . . . 38
2-23 Table creator of your CM database reflects the qualifier in your bind statements . . . 39
2-24 Invoking DB2 Admin without access to Change Management . . . 40
2-25 Invoking DB2 Admin with access to Change Management . . . 40
2-26 Choosing option 5 ‘Manage ID table’ . . . 41
2-27 Access denied on the ID table . . . 42
2-28 Identifying the CM packages . . . 43
3-1 Active DB2 Systems panel . . . 46
3-2 DB2 Admin Launchpad . . . 47
3-3 DB2 Admin Menu . . . 47
3-4 Change DB2 Admin Parameters panel . . . 48
3-5 Allocating the print data set . . . 49
3-6 Changing the disposition of the print data set . . . 49
3-7 Batch parameters panel . . . 50
3-8 Prompt Options panel . . . 51
3-9 Statement Execution Prompt . . . 52
3-10 System Catalog panel - object options . . . 53
3-11 System Catalog panel - authorization options . . . 53
3-12 Main links between DB2 catalog tables . . . 54
3-13 Searching table spaces in database DSG% and database owner PAOLOR3% . . . 55
3-14 Result of searching table spaces in databases DSG% . . . 56
3-15 Result of showing table spaces which belong to database DSG24D03 . . . 57
3-16 Display search fields for a DB2 system catalog table . . . 58

© Copyright IBM Corp. 2007. All rights reserved.

3-17 Table spaces with four tables 58
3-18 Changing the sort order 59
3-19 Table spaces ordered by active pages descending 60
3-20 Specifying table spaces to be sorted by name 61
3-21 Table spaces sorted by name - Result set 62
3-22 Table display panel 63
3-23 Showing all rows in BROWSE format 63
3-24 Show all columns of a single row using the / line command 64
3-25 Build SQL SELECT Prototype panel 65
3-26 List table spaces with more than one partition sorted by name 66
3-27 Select table spaces in BP1 that begin with GLW 67
3-28 Resulting list of table spaces with a name starting with GLW in BP1 67
3-29 Column names of SYSIBM.SYSTABLESPACE 68
3-30 Execute SQL Statements panel 69
3-31 Build SQL Prototype panel 69
3-32 Build SQL prototype options panel 69
3-33 Build SQL Select Prototype panel 70
3-34 Issuing the >PRINT TABLE command 71
3-35 Selecting columns for the print data set 71
3-36 Contents of print data set 72
3-37 Generating DDL for a table 73
3-38 Generated DDL for a table 74
3-39 SQL - Execute & Copy panel 74
3-40 Statement Execution Prompt panel 75
3-41 Change Management Prompt panel 75
3-42 Generating a database 76
3-43 Generating SQL for a database from DB2 catalog 77
3-44 Structure of new database DSG24P03 79
3-45 Switch Catalog Copy option 80
3-46 Select Copy of DB2 Catalog panel 80
3-47 System Catalog panel after switching to a catalog copy 81
3-48 Object structure from database DSG24D03 81
3-49 Display DSN that is associated with the table space, index space, or partition 82
3-50 Data set name for table space DSG24D03.GLWS001 82
3-51 VSAM cluster information for table space DSG24D03.GLWS001 83
3-52 Database authorizations for PAOLO on databases DSG24D0% 83
3-53 Revoke Database Privileges panel 84
3-54 Revoke impact report for PAOLO on database DSG24D03 84
3-55 Statement Execution Prompt panel 85
3-56 List of all table spaces in database DSG24D03 86
3-57 Drop table space GLWSPRJ 86
3-58 DROP impact analysis summary 87
3-59 Drop Impact analysis details panel 88
3-60 Manage Batch Job Checkpoint Table panel 89
3-61 Display Batch Job Checkpoint Table 89
3-62 Display Checkpoint Table 90
3-63 WSL operations 92
3-64 Specify Work Statement List 93
3-65 Manage Work Statement Lists 94
3-66 List WSL(s) in WSL Library 94
3-67 Show Work Statement List GEN00001 panel 95
3-68 Show only DDL Statements for WSL GEN00001 97
3-69 Edit a DDL Work Statement 97

3-70 Insert Work Statement 98
3-71 Edit Statement 98
3-72 Show Work Statement List GEN00003 99
3-73 Interpret WSL Options panel 100
3-74 Interpret WSL Report panel 101
3-75 Generated JCL for WSL validation 102
3-76 Validate Work Statement List report 103
3-77 Generate job for Work Statement List RSTATS01 104
3-78 Clone Work Statement List panel 105
3-79 Specify Mask Table or Data Set 106
3-80 CM Insert Mask panel 107
3-81 CM Mask Lines panel 107
3-82 Successful generation of WSL GEN0001C 109
3-83 Show Work Statement List GEN0001C 110
4-1 Change Management model before the new CM functions 114
4-2 New Change Management Process 115
4-3 CM is active as shown on the main menu and can be selected 116
4-4 Change Management main menu 117
4-5 CM- Manage ID Table entries 117
4-6 Display CM Changes panel, giving an overview of changes 118
4-7 CM Change Statements panel showing all the change statements for a change 119
4-8 Expanding the complete statement in browse mode 119
4-9 Overview on generating changes 120
4-10 Overview of process flow to promote a change into another environment 121
4-11 The content of a delta changes data set 122
4-12 Tables relationships 123
4-13 Overview of CM databases in multiple DB2 environments 124
4-14 Example of CM-Changes 125
4-15 CM Changed Objects panel - All affected objects for a change 125
4-16 Change backout with RECOVER 127
4-17 Granularity of RECOVER 128
4-18 Change Management Menu 128
4-19 Query your changes 129
4-20 The Changes summary panel 130
4-21 Show statements of a change 131
4-22 The statement summary 131
4-23 Report from changes 132
4-24 Looking up changed objects 133
4-25 Summary of states 134
4-26 The REGISTER process 135
4-27 Create a change with status INIT 136
4-28 Inserting (or creating) a change 136
4-29 Initial status of a change 137
4-30 Complete the change definition 138
4-31 The ANALYZE workflow 139
4-32 Using the ANALYZE command 140
4-33 The ANALYZE process parameter panel to generate ANALYZE JCL 141
4-34 The version methods 142
4-35 Standard WSL member names are generated during ANALYZE 144
4-36 Temporary files during ANALYZE process 145
4-37 Error message provided if DSNDB07 is not available before ANALYZE 145
4-38 Run process 146
4-39 Information about a change before entering the RN command 146

Archived

4-40 The RUN panel 147
4-41 The output shows that runtime analysis detects a mismatch 148
4-42 Restarting a change with ER 149
4-43 Modifying the RESTART step of the job 149
4-44 Taking a new base version during RUN 150
4-45 Protection status of a version 151
4-46 The PROMOTE workflow 151
4-47 Invoking the PROMOTE process 152
4-48 Where to enter your BASE versions while doing the PROMOTE process 153
4-49 A change for the PROMOTE process 154
4-50 When comparing two base versions, the corresponding change is DEFINED 154
4-51 Delta change generated DDL 155
4-52 The IMPORT workflow 156
4-53 The IMPORT screen 156
4-54 Action for pending changes 157
4-55 Using Masking while doing the IMPORT process 157
4-56 CM Recover Process 158
4-57 Generate Analyze Job panel with recover option 160
4-58 Generate Analyze Job panel with recover option 160
4-59 CM Changes panel showing status for change type CHANGE and RECOVER. . . . 161
4-60 CM Changes panel showing status after recover completed 161
4-61 CM Restart capability overview. 164
4-62 Using restart capability on the Changes panel 165
4-63 Modify change statement on the Change Statements panel 165
4-64 Change Statements panel after editing the statement 165
4-65 Choosing a mask for edit 166
4-66 Masking descriptions 167
4-67 A warning pop-up when updating a mask being used in a non-complete change . . 168
4-68 Showing the ignore lines. 169
4-69 Update an ignore field from SYSIBM.SYSTRIGGERS 170
4-70 Preventing the trigger OWNER from comparing. 170
4-71 OWNER has been added to SYSTRIGGERS for ignoring 171
4-72 Using ALC on a DSN8 sample table. 172
4-73 Asking for Change Management 173
4-74 Using ALC. 173
4-75 Enter your change. 174
4-76 Register successful. 174
4-77 Manage changes 175
4-78 Display changes 175
4-79 Change Panel according to your query. 176
4-80 Detecting an existing pending change 177
4-81 Diagram Detecting Pending Changes. 177
4-82 Enter the command CH to display your change entry 178
4-83 Enter the command SR to see the related objects 179
4-84 The related objects 179
4-85 Change interpretation 180
4-86 Execute SQL. 181
4-87 Execute SQL statements from screen input 181
4-88 Change Management 182
4-89 Possibility to run an emergency change 182
4-90 Showing pending changes before doing an emergency change 183
4-91 Emergency change has been executed 183
4-92 Adding a column 184

4-93 Change Management . . . 184
4-94 Register change . . . 184
4-95 Fast change is completed . . . 185
4-96 Look up for prerequisite changes . . . 185
4-97 Prereq changes can easily be determined by typing PQ . . . 186
4-98 Build a base version during the RUN process . . . 187
4-99 CM - Manage versions option . . . 188
4-100 CM - Manage Versions panel . . . 188
4-101 CM - Insert Version Scope . . . 189
4-102 Manage Versions panel - Display version scopes . . . 189
4-103 CM - Version Scopes . . . 189
4-104 Example of version scope definition . . . 190
4-105 CM - Generate version . . . 191
4-106 CM - New base version . . . 191
4-107 JCL to create Base Version (1/2) . . . 192
4-108 JCL to create Base Version (2/2) . . . 193
4-109 Versions panel showing a base version . . . 194
4-110 Build and store delta versions in CM database during REGISTER process . . . 194
4-111 Use of versions in the ANALYZE process . . . 195
4-112 CM Versions - The whole picture . . . 196
4-113 Implementing changes and rolling out new version (COMPARE) . . . 197
4-114 Implementing changes and rolling out new version (ALC) . . . 198
5-1 The database environment . . . 202
5-2 Creating database DSG25D0P based on existing database DSG24D02 . . . 205
5-3 Generating a database . . . 205
5-4 Generating SQL for a database from the DB2 Catalog . . . 206
5-5 Specify Mask Data Set panel . . . 208
5-6 Insert Mask panel . . . 208
5-7 Execution of Insert Mask . . . 208
5-8 Masking entries (Type, From, and To) provided by the user . . . 209
5-9 CM Mask Lines entry panel . . . 209
5-10 Specify Work Statement List prompt panel . . . 210
5-11 Issue the WSL command to get to the Work Statement List panel . . . 210
5-12 Structure of new database DSG25D0P with changed table space name . . . 211
5-13 Showing the available options to clone DDL with data . . . 212
5-14 Clone with MIGRATE within the same DB2 subsystem . . . 213
5-15 System Catalog panel . . . 214
5-16 Selecting Migration source database . . . 215
5-17 Selecting migration for all table spaces . . . 215
5-18 Migrate Parameters panel 1/3 . . . 216
5-19 Migrate Parameters panel 2/3 . . . 217
5-20 Migrate Parameters panel 3/3 . . . 218
5-21 Specify Work Statement List panel . . . 219
5-22 Migrate Progress panel . . . 219
5-23 Migrate Jobs . . . 219
5-24 Job flow for cloning a database within same DB2 subsystem . . . 220
5-25 Clone into another DB2 subsystem . . . 222
5-26 Migrate Parameters panel for remote target system 1/2 . . . 223
5-27 Migrate Parameters panel for remote target system 2/2 . . . 224
5-28 Migrate Parameters panel 3/3 . . . 225
5-29 Specify Work Statement List panel . . . 226
5-30 Migrate Progress panel . . . 226
5-31 Migrate Jobs . . . 226

Archived

5-32 Migrate and Compare technique to clone multiple databases 227
5-33 Selecting migration source databases 229
5-34 Migrate - Add function 229
5-35 Add more objects to this migrate operation 230
5-36 Migrate Add Database panel 230
5-37 Selecting objects to be added to the migration list 231
5-38 Additional objects in the migration list 231
5-39 Scenario for cloning multiple databases with MIGRATE. 232
5-40 Migrate Parameters panel (1/2) 233
5-41 Migrate Parameters panel (2/2) 233
5-42 Migrate batch jobs 234
5-43 CM Mask Lines panel 234
5-44 Work Statement List Edit mode - Unload data set names 235
5-45 Work Statement List Edit mode - Data set names and table space names 235
5-46 WSL Edit mode - LOAD statement with unloaded data set names and table spaces 236
5-47 Cloning with Compare against empty target 237
5-48 DB2 Compare Menu 237
5-49 Specify DB2 Source Catalog Extract 238
5-50 Specify Source DB2 Databases panel 238
5-51 Compare Add Databases panel 1/2 238
5-52 Compare Add Databases panel 2/2 239
5-53 Compare Add Databases - after selection 239
5-54 Specify Source DB2 Databases - After selection 239
5-55 Specify DB2 Target Catalog Extract 240
5-56 Specify DB2 Target Databases. 240
5-57 Specify Compare Masks 241
5-58 Specify CM Masks and select with “+” 241
5-59 CM Mask Lines panel 241
5-60 Specify Compare Ignore Fields lookup with “?” 242
5-61 CM Ignores can be selected with “+” 242
5-62 Specify Ignore Fields panel. 243
5-63 Generate Compare Jobs panel 1/3 244
5-64 Generate Compare Jobs panel 2/3 244
5-65 Generate Compare Jobs panel 3/3 245
5-66 Change Management Prompt panel 245
5-67 Change Management Prompt panel 245
5-68 Specify Register Mode 246
5-69 Specify Register Mode 247
5-70 Display CM Changes panel 247
5-71 Cloning multiple databases with new CM function 248
5-72 Overview on import to process delta or DDL files 249
5-73 CM Changes panel 249
5-74 Import Changes panel. 250
5-75 Import a Change - Action for Pending Changes panel 250
5-76 Register Options panel 251
5-77 Joblog of failed IMPORT 251
5-78 Pending Changes 251
5-79 Changes panel 252
5-80 Show related objects for the change 252
5-81 Import a Change - Action for Pending Changes panel 253
5-82 Changes panel 253
5-83 Changes panel 254
5-84 Run a change panel 254

5-85 Databases panel showing the new databases 255
5-86 Register the import with mask. 255
5-87 Mask lines panel 256
6-1 CM Version panel (ADB2C41) 258
6-2 CM Interpreting of a Version panel (ADB2C41I) 259
6-3 CM Version Scope panel (ADB2C42) 259
6-4 CM - Version Scope Objects (ADB2C4O). 260
6-5 DB2 Administration Menu 7.2.0 (ADB2) 261
6-6 System Catalog (ADB21) 262
6-7 Tables, Views, and Aliases panel (ADB21T) 263
6-8 Alter Table panel (ADB21TA. 264
6-9 Alter Table panel (ADB21TAB) 265
6-10 Change Management Prompt (ADB2CMPR) 265
6-11 CM - Register Change (ADB2CMRG). 266
6-12 CM - Pending Changes (ADB2CMPC) 267
6-13 CM - Register Options (ADB2CRO) 267
6-14 CM - Register Options completed successfully. 268
6-15 CM - Changes (ADB2C11) 268
6-16 Changes (ADB2C11) 269
6-17 Generate Analyze Job (ADB2C11A). 270
6-18 Run a Change (ADB2CEX1). 271
6-19 Versions (ADB2C41). 271
6-20 Changes (ADB2C11) 272
6-21 Run a Change (ADB2CEX1). 272
6-22 Changes (ADB2C11) 273
6-23 Tables, Views, and Aliases (ADB21T) 273
6-24 Change Management Prompt (ADB2CMRO) 274
6-25 Alter Table (ADB27C) 274
6-26 Alter Table (ADB27C) 275
6-27 Create Table Column Number 1 (ADB26CTU). 276
6-28 Create Table Column Number 1 (ADB25CTU). 277
6-29 DB2 Admin ALTER Column Truncation (ADB27CT) 277
6-30 Alter Table (ADB27C) 278
6-31 Alter Tables (ADB27CA) 279
6-32 Redefine Index (ADB21XAR) 279
6-33 Clustering Index Confirmation (ADB2CONF) 280
6-34 Redefine Index - Space (ADB21XAS) 280
6-35 ISREDDE2 281
6-36 Alter Tables (ADB27CA) 281
6-37 Register Options (ADB2CRO) 282
6-38 Register Options (ADB2CRO) 282
6-39 Changes (ADB2C11) 283
6-40 Generate Analyze Job (ADB2C11A). 284
6-41 Setting the recovery job name 285
6-42 Job output 285
6-43 Changes in ANALYZED Status. 286
6-44 Run a Change (ADB2CEX1). 286
6-45 Version Scopes (ADB2C42) 287
6-46 Run a Change (ADB2CEX1). 287
6-47 Table Space (ADB21S). 288
6-48 Alter Table Space (ADB21SA) 289
6-49 Redefine Table Space (ADB21SAR) 289
6-50 Redefine Table Space (ADB21SAR) 290

6-51 Create Partition (ADB21SAX) 291
6-52 Redefine Table Space (ADB21SAR) 291
6-53 Administration Menu 7.2.0 (ADB2) 294
6-54 DB2 Object Comparison Tool Menu (GOCMENU) 294
6-55 Specify Compare Source (GOC1) 295
6-56 Specify Source DDL File (GOC11) 295
6-57 Specify Compare Target (GOC1) 296
6-58 Specify Target DB2 Location (GOC14) 296
6-59 DB2 Object Comparison Tool Menu (GOCMENU) 297
6-60 Generate Compare Jobs (GOC5) 298
6-61 Change Management Prompt (ADB2CMRO) 299
6-62 CM - Register Options (ADB2CRO) 299
6-63 Specify Register Mode (GOC5RM) 299
6-64 job output 300
6-65 Changes (ADB2C11) 301
6-66 Change Management (CM) (ADB2C) 303
6-67 CM - Manage Ignores (ADB2C3) 303
6-68 CM - Insert Ignore (ADB2C22) 304
6-69 CM - Ignores (ADB2C31) 304
6-70 Specify Ignore Fields: Objects (GOCCI) 305
6-71 Change Management (CM) (ADB2c) 306
6-72 CM - Manage Masks (ADB2C2) 307
6-73 CM - Insert Mask (ADB2C22) 307
6-74 CM - Insert Mask (ADB2C22) 307
6-75 CM - Masks (ADB2C21) 308
6-76 Mask Lines (ADB2C2L) 308
6-77 CM - Register Options (ADB2CRO) 309
7-1 Rolling out using PROMOTE 312
7-2 Manage versions 313
7-3 Looking up your version scope 313
7-4 Looking up the versions for a version scope 314
7-5 Base versions 314
7-6 The PROMOTE process 315
7-7 Register a change for the PROMOTE process 315
7-8 The status of a change after doing the PROMOTE process 316
7-9 Information about the PROMOTE process 316
7-10 Content of the delta changes data set 317
7-11 Rolling out using DB2 Compare 317
7-12 Invoking DB2 Compare 318
7-13 DB2 Compare Main Menu 319
7-14 Specifying the source 319
7-15 The source is table 320
7-16 Specifying the table name 320
7-17 Specifying the compare target 321
7-18 Specifying a mask 321
7-19 The current definition of mask PAOLOR1.MASTERMASK 322
7-20 The DB2 Compare parameter panel 323
7-21 Using Change Management 324
7-22 Change registration with Compare 324
7-23 Error message if the DB2 Compare job is submitted twice 325
7-24 The changes DD statement 325
7-25 DB2 Compare delta changes data set 326
7-26 The used change during the compare process 326

7-27 Importing a delta change data set. 327
7-28 Register of your IMPORT 327
7-29 Status of your imported change 328
7-30 Change statements of your change 328
8-1 CM Recover Process in detail. 330
8-2 Recovery of a completed change with existing data 332
8-3 Version Scopes panel 333
8-4 Version Scope Objects 333
8-5 Versions panel 334
8-6 ALTER table with warning in upper right. 334
8-7 Register Options 335
8-8 Register Options panel with successful register message 335
8-9 Changes panel 336
8-10 Change Statements panel 336
8-11 Changes panel 337
8-12 Generate Analyze Job panel. 338
8-13 Register Options panel for RECOVER 338
8-14 Generated analysis Job 339
8-15 Changes panel showing ANALYZED status for RECOVER and CHANGE type . . . 339
8-16 Work Statement List Library panel 340
8-17 Changes panel 340
8-18 Run a change panel 341
8-19 Generated Job for the change 342
8-20 CM Changes panel showing the change in COMPLETE status 342
8-21 Status of changes 343
8-22 CM Changes panel and issuing the RC (recover) command 343
8-23 Recovery Strategy panel showing pending changes 344
8-24 Generated recover JCL with PCACT PARM. 345
8-25 Changes panel showing recover in running status 345
8-26 Changes panel showing recover in complete status. 346
8-27 Columns of recovered table 346
8-28 Recover a completed change with original data 347
8-29 Changes panel 347
8-30 Changes Statements panel showing all registered changes 348
8-31 Generate Analyze Job panel with recover to original data 348
8-32 Register Options panel for recover change 349
8-33 Work statement list showing the unload data set name 350
8-34 Work statement list for the recover change using the original data set. 350
8-35 Recover Work statement list content for UTL UTILFROM line 351
8-36 Data set generated by change and to be used from recover change 351
8-37 Changes panel showing that the CHANGE is COMPLETE 352
8-38 Generated recovery JCL. 352
8-39 Recover JCL with recover work statement list inside the job 353
8-40 CM Changes panel showing the recover in COMPLETE status. 354
8-41 Recover to existing data after change abnormally terminated 354
8-42 Changes panel - Original change in status RUNNING and recover change in status ANALYZED. 355
8-43 Failed CM run job for change Id 228. 356
8-44 Change Management recover job (1/2) 357
8-45 Change Management recover job (2/2) 358
8-46 Changes panel - Recover change in status COMPLETE 358
8-47 Changes panel - Original and recover change in status CANCELED. 359
8-48 Backout to a previous base version 359

Archived

8-49 Promote panel - create delta file 360
8-50 Register options panel for compare change 360
8-51 Changes panel - Compare change in CAN. 361
8-52 Import Changes panel - Specify input data set 361
8-53 Register Options panel for promote change 361
8-54 Changes panel - promote change in status defined 362
8-55 Change statement taken from the change statement table 362
8-56 Extract of work statement list content showing the DDL part 362
8-57 Change statement taken from the change statement table for the recover change . 363
8-58 Extract of work statement list content showing the DDL part 363
9-1 Sending Doc to FTP, II11945 369
9-2 System Parameters (ADB2Z2Z) 370
9-3 Debug Options (ADBDEBUG) 370
9-4 System Parameters (ADB2Z2Z) 371
9-5 Current Data Set Allocations using TSO ISRDDN 372
9-6 Allocating a new ADBDIAG. 372
9-7 ISPF Command Entry Panel. 373
9-8 Generate Compare Jobs panel (GOC5) 375
9-9 WSL job output with Data conversion errors. 376
9-10 ALTER Table (ADB27C) 377
9-11 Alter Tables (ADB27CA) 378
9-12 JCL example of DROP-SC 379
9-13 Alter Table (ADB27C) 380
9-14 Alter Tables (ADB27CA) 380
9-15 JCL example of DROP 381

Archived

Tables

2-1 Table space allocation 17
2-2 Summary of installation steps 30
3-1 Performance considerations 64
4-1 CM data base tables 122
4-2 List of available commands for change. 130
4-3 Processes and link to description 134
4-4 Change action for table columns and resulting data recover generations 159
4-5 Status of changes 162
4-6 Type of changes 163
6-1 Physical Changes 293
6-2 Logical Changes 293
6-3 The DB2 catalog table ignore fields 302
9-1 Matrix for data type and length changes that do not require data conversion 374

Archived

© Copyright IBM Corp. 2007. All rights reserved.

Archived

Examples

2-1 A runtime environment for DB2 Admin and DB2 Compare 32
2-2 Alias names referring to your runtime environment 33
3-1 Update userid in checkpoint table 90
3-2 Job to execute WSL TEST01A 104
4-1 ALTER statement represents your change 137
4-2 The keyword values of the request parameters for each object type 190
4-3 DDL delta 194

Archived

© Copyright IBM Corp. 2007. All rights reserved.

Archived

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

© Copyright IBM Corp. 2007. All rights reserved.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

DB2®	MVS™	Redbooks (logo) ®
IBM®	OMEGAMON®	REXX™
IMS™	RACF®	VTAM®
Language Environment®	Redbooks®	z/OS®

Other company, product, or service names may be trademarks or service marks of others.

Archived

Preface

A database administrator (DBA) has a challenging job trying to maintain databases in today's nonstop 24x7 environments. This challenge is compounded by the complexity of database structures and the time pressure of trying to be up again as soon as possible without making mistakes.

When business expands, several DB2® subsystems have to be maintained, such as multiple development systems, function test, acceptance test, volume test, and production systems. Often changes have to be implemented across multiple DB2 subsystems.

The two IBM® DB2 for z/OS® tools that help database administrators to implement database changes are the DB2 Administration Tool and the DB2 Object Comparison Tool.

In this IBM Redbooks® publication we show how, with Version 7.2, the two tools offer an enhanced Change Management function to manage and track the changes of your DB2 objects. We show how the tools help you define changes, resolve conflicts, register the changes, analyze the changes to generate a work statement list that applies the changes, and then run the changes in the correct order. The Change Management functions also allow you to back out completed changes and facilitate the management of a convenient audit trail.

IBM DB2 Administration Tool for z/OS and DB2 Object Comparison Tool have been described in two previous IBM Redbooks: *DB2 for z/OS Tools for Database Administration and Change Management*, SG24-6420, published in July 2003, and *Administration Solutions for DB2 UDB for z/OS*, SG24-6685-00, published in April 2005. In those books, these tools are at the V4.2 and V5.1 levels, respectively.

More specifically, the first book contains a general description of most functions offered by the two tools.

In the second book, Chapter 6. "Using DB2 Tools with DB2 UDB for z/OS Version 8" describes the main functions that are available with the DB2 Tools when working in a DB2 UDB for z/OS Version 8 environment.

The tools described in this book are at the V7.2 level, including all maintenance available at end of March 2007. This book concentrates on the new Change Management functions.

For details on functions and the chronology of functions by release, refer to the section "Summary of Changes" in *IBM DB2 Administration Tool for z/OS User's Guide and Reference Version 7 Release 2,* SC18-9552.

The team that wrote this IBM Redbooks publication

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.

Paolo Bruni is a DB2 Information Management Project Leader at the International Technical Support Organization, San Jose Center. In this capacity he has authored several IBM Redbooks on DB2 for z/OS and Data Management tools, and has conducted workshops and seminars worldwide. During Paolo's many years with IBM, in development and in the field, his work has been mostly related to database systems. Before joining the ITSO in 1998, Paolo was a Certified Consultant IT Architect with IBM Italy.

© Copyright IBM Corp. 2007. All rights reserved.

Lucy Krantz is a DB2 Tools Admin and Object Compare Lv2, Advisory Software Engineer, San Jose, California. Before Data Management tools she was Lv2 Support in DB2 with expertise in IRLM, Data Sharing, and DB2 Systems. During Lucy's many years with IBM, she has worked as technical customer support in various areas involving database systems and before that was a hardware Customer Engineer in San Diego, CA.

Frank Nikodem is a DB2 system engineer with Credit Suisse in Zurich, Switzerland. Before joining Credit Suisse in 2003, Frank worked as a DB2 Specialist for a data center in Germany. Frank has been working with DB2 for z/OS for the last 9 years. He has been responsible for DB2 installation and migration, as well as for third party vendor tools. He is now in charge of DB2 related tasks such as logging infrastructure, security, catalog/directory and performance analysis in a large data sharing environment. He also provides guidance for the utilization of DB2 Data Management tools. Frank has been recently concentrating on the Change Management process.

Guenter Pfann is a Senior IT Specialist working for IBM Switzerland with DB2 Tools on z/Series. He joined IBM in 2002, after working 10 years with BMC as technical DB2 tools specialist. He has 15 years experience in DB2 Change Management. Before that he worked as Systems Engineer, DB2 consultant and DBA at various customers. He works with DB2 since 1987 and started in 1980 on MVS™ as application programmer with IMS™ DB/DC and VSAM.

Marc White is working as a Technical Consultant and Senior Accredited IT Specialist for IBM Software Services in Zurich, Switzerland. He has several years of experience with DB2, primarily on the z/OS platform. His areas of expertise include DB2 data sharing, high availability, performance analysis, diagnosis, and solution of problems relating to the Database Management System and associated Software. Marc has co-authored the IBM Redbooks publication, *DB2 for z/OS Tools for Database Administration and Change Management*, SG24-6420-01.

Archived

Here we show a photo of the team.

Left to right: Frank, Lucy, Guenter, Paolo, and Marc

Thanks to the following people for their contributions to this project:

Rich Conway
Bob Haimowitz
Emma Jacobs
Deanna Polm
Sangam Racherla
Yvonne Lyon
International Technical Support Organization

Ty Bemis
John Dembinski
Torben Flygenring
Roger Hong
Tom Ulveman Jensen
Vijaya Katikireddy
Jim Kennelly
June Kin
Elaine Morelli
Mary Petras
Tom Ramey
Joseph Reynolds
Richard Schaufuss
David Schwartz
Joe Sinnott

Archived

Ruth Tekus
Patti Tonello
Sofilina Wilhite
IBM Silicon Valley Lab, San Jose

Mike Bracey
Steve Speller
IBM UK

Thanks to the authors of the previous IBM Redbooks publication, *DB2 for z/OS Tools for Database Administration and Change Management*, SG24-6420, published in July 2003:

Alan Black
Bob Stone
Marc White

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbooks publication dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our IBM Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Archived

Part 1

Introduction

In this part of the book, we introduce the two tools: DB2 Admin and Object Compare. The reader already familiar with the products can go directly to Part 2, “Getting started with Change Management” on page 13.

This part consists of just one chapter:

- Chapter 1, “Tools overview” on page 3

Archived

© Copyright IBM Corp. 2007. All rights reserved.

Archived

1

Tools overview

In this chapter we provide a brief overview of the two products that make up this book, the IBM DB2 Administration Tool for z/OS and the IBM DB2 Object Comparison Tool for z/OS.

We also describe how the two products together address most of the issues related to a database Change Management process.

We cover the following topics:

- Products: Overview
- Change Management support

Archived

© Copyright IBM Corp. 2007. All rights reserved.

1.1 Products: Overview

This section gives a high level overview of:

- DB2 Administration Tool for z/OS
- DB2 Object Comparison Tool for z/OS

In this book we do not attempt to give a detailed description of all the function offered by the two products. Especially with DB2 Administration Tool for z/OS, this would be a challenge. Rather, we focus on those functions that database administrators use in their daily job when managing DB2 objects, such as creating, changing, and migrating. For details on the other functions, refer to the documentation listed in “Related publications” on page 383.

1.1.1 DB2 Administration Tool for z/OS

DB2 Administration Tool for z/OS, or *DB2 Admin* as we refer to it throughout this book, provides a comprehensive set of functions to help you manage your DB2 efficiently and effectively, even though your IT environment is complex. It offers a comprehensive set of database administration functions and manages changes to DB2 objects.

DB2 Admin provides in-depth catalog navigation by displaying and interpreting objects in the DB2 catalog and executing dynamic SQL statements. It is integrated with other DB2 utilities to simplify the creation of DB2 utility jobs, which creates additional functionality with product-specific line commands for table editing, SQL cost analysis, and path check analysis.

Figure 1-1 describes the major DB2 Admin functions.

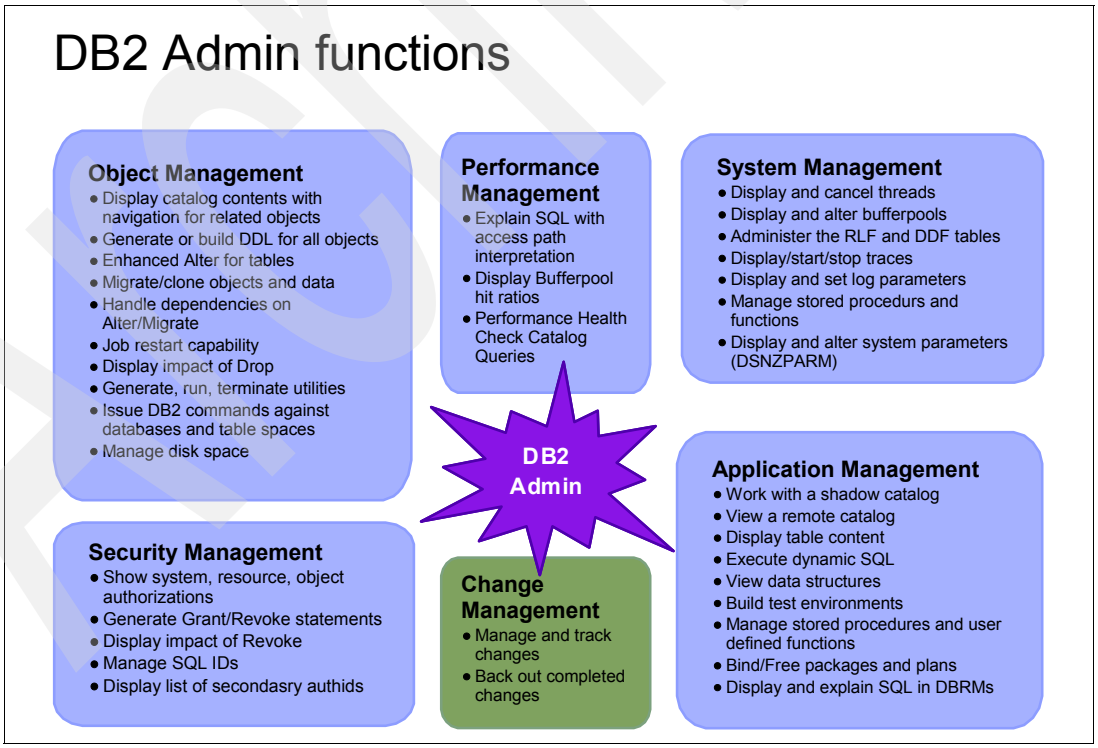

Figure 1-1 DB2 Admin Tool overview

From the DB2 Admin panels you can issue primary commands and line commands. Primary commands can be issued from the command line on the panel and their syntax is explained in the help panels. Line commands are issued from ISPF table display panels and specify an operation that is to be performed on the information that is displayed on the row. You also can define your own line commands during installation.

Object Management

Because DB2 Admin provides in-depth DB2 catalog navigation, it minimizes the time required to review the catalog. Objects in the catalog are displayed and interpreted, and relevant catalog information is presented logically. You can issue any DB2 command, including BIND, REBIND, and FREE selected plans and packages.

DB2 Admin is integrated with DB2 utilities to simplify the creation of DB2 utility jobs. JCL can be generated for DB2 utilities and running them in batch. Utility statement can also be included in a work statement list (WSL) and to be run at another time or on another subsystem. Using LISTDEFs and TEMPLATEs are also supported.

The DB2 Admin ALTER and MIGRATE functions can simplify administration tasks. After using the ALTER function to specify desired changes, the tool generates the jobs required to implement these changes. These jobs unload the data, recreate the table and reload the data. The tool handles all object dependencies during ALTER and MIGRATE. And, after the MIGRATE function has defined the target DB2 subsystem, DB2 Admin creates the jobs required to copy definitions and data to the target:

- The ALTER function lets you change the name and attributes of a table or column, insert new columns and drop existing columns.
- The MIGRATE function facilitates the copying of all the objects and data in one or more databases or table spaces to another DB2 subsystem.

Prompt options can be activated for five types of statements; definition, authorization and update SQL, DB2 commands and DSN commands. These options enable you to edit or execute the statements, put them in work statements, or run them in batch jobs.

ALTER of primary key characteristics can be propagated to foreign keys.

DROP impact analysis is provided to prevent inadvertent data loss when dropping tables.

DB2 Admin also helps you manage disk space by gathering VSAM related information in a convenient way, and also helps you with space estimation of your DB2 objects.

Security Management

DB2 Admin allows you to display authorizations that have been granted on any type of DB2 object, and enables you to REVOKE these authorizations or GRANT new ones.

The tool also provides REVOKE impact analysis to prevent inadvertent data loss when revoking authorities.

It allows you to display the list of secondary authorization IDs, as well as to change your SQLID.

Performance Management

DB2 Admin has a built-in EXPLAIN function that allows you to EXPLAIN a query and provides an interpretation of the PLAN_TABLE output into English sentences.

The tool also comes with a set of performance health check catalog queries.

Change Management

The Change Management (CM) function can be used to manage and track the changes to the DB2 objects. It allows groups of users to collaborate in building changes by managing information through a series of DB2 tables stored in an single database. Making changes through the DB2 Admin Change Management process provides a convenient audit trail like determination where the changes are in the process, and understand which changes have been deployed.

Completed changes that have been made through Change Management can be recovered (one at a time).

System Management

DB2 Admin allows you to display and cancel threads, display and alter buffer pools, and display, start and stop DB2 traces, set and display the logs.

It also provides a very convenient way to administer RLF and DDF tables.

DB2 Admin manages stored procedure operations, like creating, displaying or altering, issues the DB2 START and STOP STORED PROCEDURE command and showing statistics for stored procedures that are accessed by DB2 applications. Other system administration functions like managing functions are also supported.

Dynamic DSNZPARM changes enable you to display current DSNZPARMs, change parameters, generate new DSNZPARM modules with changes and activate those changes in DB2.

Application Management

DB2 Admin allows you to work with a copy of the DB2 catalog to avoid contention and other performance problems on the actual catalog.

You can access a remote DB2 catalog, as long as a DDF connection exists between the systems. This enables you to do centralized management of all your DB2 subsystem through a single DB2 Admin session.

You can also execute any dynamic SQL statement through DB2 Admin, or you can invoke SPUFI from within DB2 Admin.

The integration of DB2 Admin with other DB2 tools creates additional functionality with product-specific line commands for table editing, SQL cost analysis and path check analysis. Through its DB2 tools launch pad, DB2 Administration Tool for z/OS is the central access point for the current tools with an interactive system productivity facility (ISPF) interface, such as DB2 Table Editor, DB2 SQL Performance Analyzer, and DB2 High Performance Unload.

1.1.2 DB2 Object Comparison Tool for z/OS

When you rely on IBM DB2 for z/OS as your production database for business-critical information, it makes sense to have a duplicate system for testing and development. But keeping these two systems as mirror images can be challenging because new applications, application modifications or errors can trigger changes in the attributes of DB2 objects in one system but possibly not in the other.

This is where IBM DB2 Object Comparison Tool for z/OS, called *DB2 Compare* in this book, can help you out. With a rich set of functions, DB2 Compare makes it easy to compare existing objects (and dependent objects) from different sources and subsequently synchronize these sources by staging and propagating changes between the environments. It maximizes system availability by keeping both database environments running in parallel.

Figure 1-2 gives you an overview of the DB2 Compare functions.

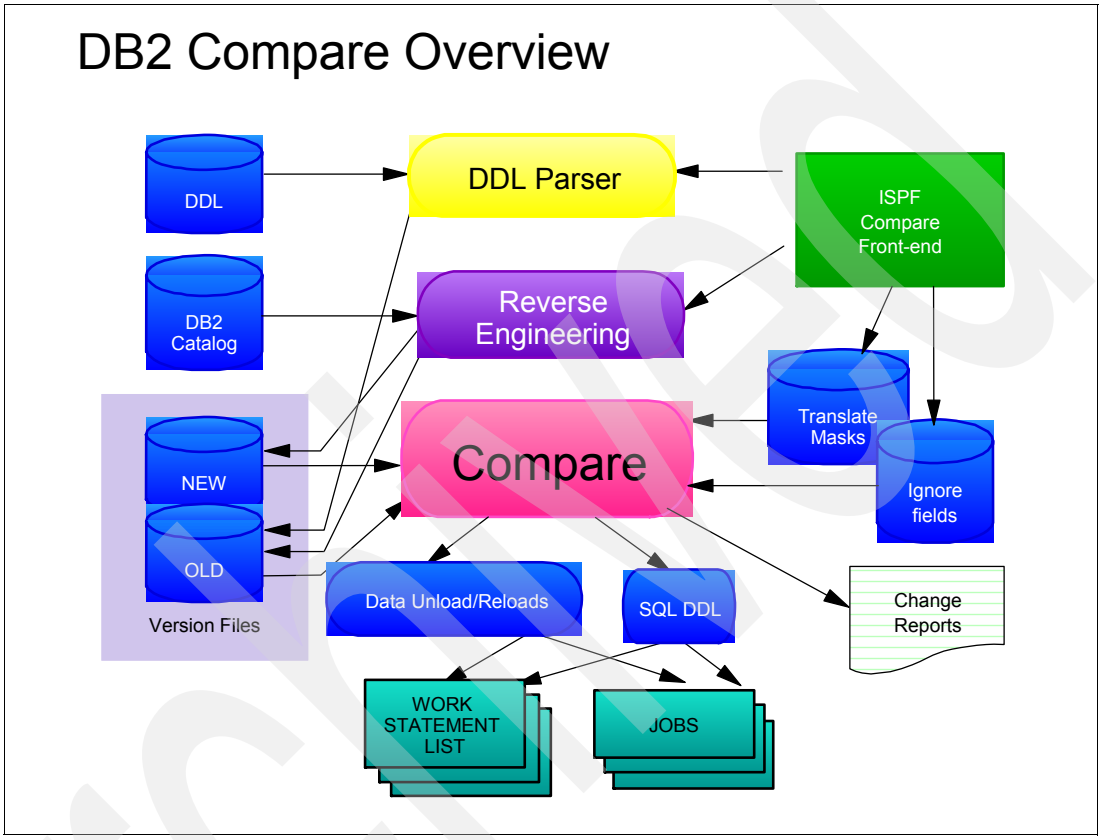

Figure 1-2 DB2 Compare overview

DB2 Compare consists of:

- An interactive system productivity facility (ISPF) front end for specification of the objects to be compared
- A DB2 catalog extract function that pulls definitions from the catalog into a version file to support the comparison process
- A GEN function which, in addition to extracting the DDL for objects, can also generate the DCL for all authorizations to the objects and the DML for the catalog statistics for the objects
- A DDL extract function that reads DDL statements and converts them into a version file
- A batch compare function that compares two version files, produces several reports describing the differences found, and generates the information required to apply changes to the target
- A batch job generator that provides everything necessary to apply the changes to the target

Archived

DB2 Compare enables you to keeps the object definitions in your DB2 test and production catalogs synchronized.

The tool's *masking* and *ignore* files, which address intentional differences and/or naming conventions, allow you to compare only the real differences that might exist between objects.

Using DB2 Compare, you can compare:

- Objects from a DB2 catalog to objects residing in another DB2 catalog
- A file with data definition language (DDL) to another file with DDL
- A file with DDL to objects in a DB2 catalog.

Whether you specify objects at the database level, the table space level or the table level, all dependent objects, such as views and indexes, are included in the comparison. Once you select an object, that object and its dependents are extracted from the catalog or DDL file and placed in a sequential data set, creating a *version* file (the source). You can then select the object to which the source should be compared. This object is extracted from its catalog and placed in a separate sequential data set creating another version file (the target).

A comparison of the source and target data sets results in the creation of a difference file. DB2 Compare generates a report of these differences for your review. Reports can be selected by the DBAs in order to choose the information included. Upon reviewing the report, you can direct the DDL generated for target changes to apply jobs. Apply jobs can be stored in a work statement list or a PDS file and then propagated to several remote sites. Changes in a test environment can be automatically migrated to the development or production environment.

DB2 Compare runs as an extension of DB2 Admin. This enables you to take advantage of the functions in DB2 Administration Tool for z/OS.

1.2 Change Management support

DB2 Admin, used in conjunction with DB2 Compare, offers a solution for handling complex processes associated with Change Management. How to work with the provided comprehensive Change Management process is illustrated later in this book using different scenarios starting with 4.2, “Overview of Change Management” on page 113.

1.2.1 Change Management in general

The goal of Change Management is to ensure that standardized methods and procedures are used for efficient handling of all changes, in order to minimize the impact of change-related incidents. Therefore an effective planned and implemented Change Management strategy is an absolute requirement.

Change Management process

The Change Management process can generally be divided into the following six steps. When taken successfully, they manage a change in an IT environment:

1. Identify potential change
2. Analyze change request
3. Evaluate change
4. Plan change
5. Implement change
6. Review and close change

The Change Management process sequence is detailed in Figure 1-3.

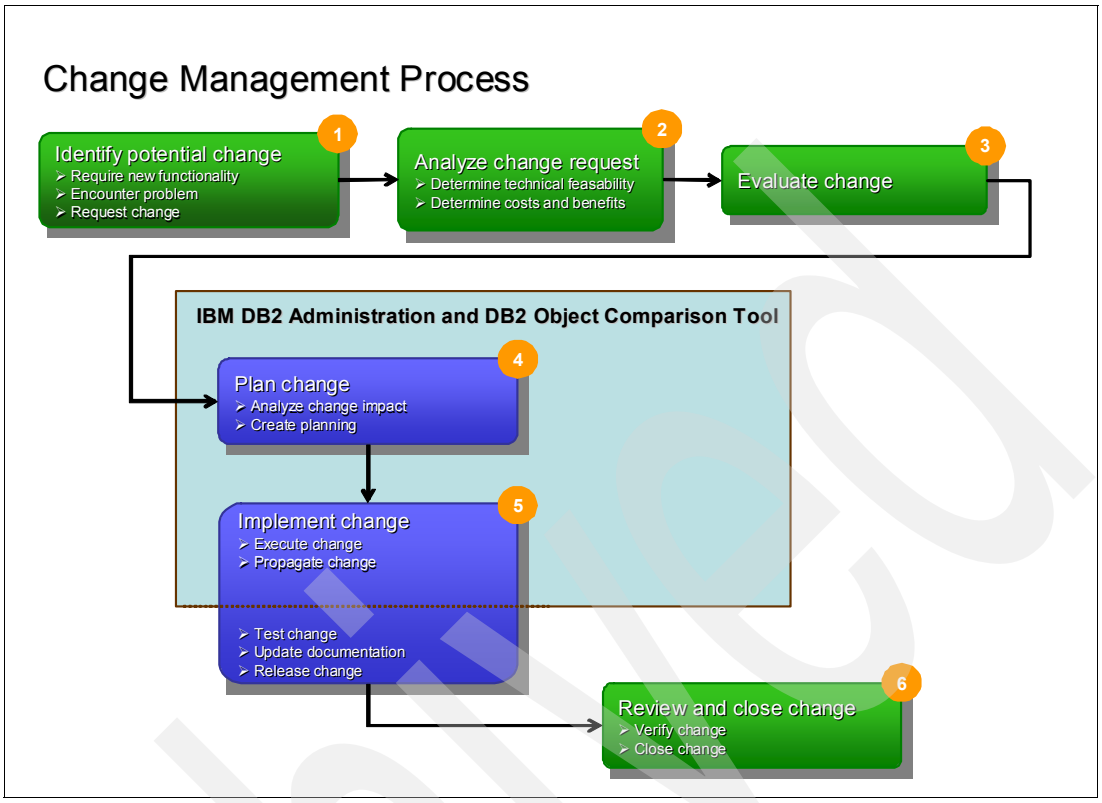

Figure 1-3 Change Management process

1.2.2 Change Management using DB2 Admin and DB2 Compare

If there is a requirement to enable a new functionality, a database structure change is required. After the technical feasibility and evaluation are complete, the Database Administrator begins to plan and execute this change.

Changes can have cascading effects on database and application performance, pushing errors deeper into the database and making it more difficult to correct mistakes. To avoid application failure, data loss or broken dependencies, DBAs must proceed meticulously through the processes of data unloading, object dropping and rebuilding, and data reloading.

DB2 Admin and DB2 Compare can eliminate the necessity to undergo such cumbersome manual object management processes.

1.2.3 Definition of new Change Management functionality

Change Management processing is the main enhancement of release 7.2 of DB2 Admin and DB2 Compare. By providing an interface to manage all the version files for a database, the tools allow you to track the history of changes as well as select version files from a list rather than having to remember them in various panels.

You are given the option to specify a change owner for every piece of work, whether you are using Alter table columns with DB2 Admin ALTER (ALC), Alter TABLE (AL), or Object Compare (OC) to effect the change. Change owner and name can then be used to provide a method for maintaining a relationship between version files, mask files, ignore files, generated DDL, and unloaded data. Using a management interface, you can track and query changes to objects and quickly find all the components involved in a change.

Changes are stored in a Change Management database (CM database), which supports the capability to report on the contents of the data and answer questions such as: Who did what, and when?

The CM function in DB2 Admin also supports collaborative processing, allowing groups of users to work together and build changes by managing information through a series of DB2 tables. The key to this process is the capability of the tool to capture only the change information, not all the related objects.

The tools can handle urgent changes that must occur immediately. They also support virtual changes for the ALC, RDEF, CREATE and RENAME database functions, giving you the ability to see changes that are in process but not apparent in the DB2 catalog. This collaborative processing ability can reduce the time you take to perform Change Management activities and decrease errors that might occur in a complex process.

The DB2 Admin CM processes are as follows:

- Register a change in the database.
- Analyze how a change modifies existing objects and generate a Work Statement List (WSL) to execute the change.
- Apply (run) a change.
- Promote (move) changes from a source system to a target system.
- Recover the changes executed.
- Query the changes that have been defined or the objects that are being changed.

1.2.4 Installation considerations and prerequisites for using CM

For each DB2 subsystem for which you plan to use the enhanced Change Management function, you must:

- Install DB2 Compare Version 7 Release 2
- Customize DB2 Admin Version 7 Release 2 so that Change Management is enabled:
  - Create CM database that stores CM relevant information.
  - Define a qualifier for the database objects in the :cmown. tag when running the ADB2CUST EXEC.
  - Bind CM relevant packages (ADBC%)

Chapter 2, “Customization” on page 15 provides detailed information about CM in terms of product tailoring. The *DB2 Administration Tool for z/OS User's Guide*, SC18-9552 contains the step-by-step procedures to activate DB2 Admin (with or without CM functionality) after completing the installation.

1.2.5 CM mandatory, optional, or not activated

The Change Management level determines whether changes must be made through Change Management. The CM levels are:

- REQUIRED:

  All changes must be registered in the Change Management database.

- OPTIONAL:

  An ISPF panel prompt appears, based on the current SQLID, where you can optionally specify to make the change through Change Management or not. This can be useful for testing CM or when that SQL ID has to make changes also outside of CM.

- NONE:

  No changes can be registered in the Change Management database.

The Change Management default level setting is NONE. Change Management levels for specific SQLIDs are stored in the <OWNER>.ADBCID table.

1.2.6 Core functions for managing DB2 object changes

For reference, Figure 1-4 illustrates the core functions in DB2 Admin and DB2 Compare for managing changes to DB2 objects prior to the Change Management functionality that was added in Version 7.2.

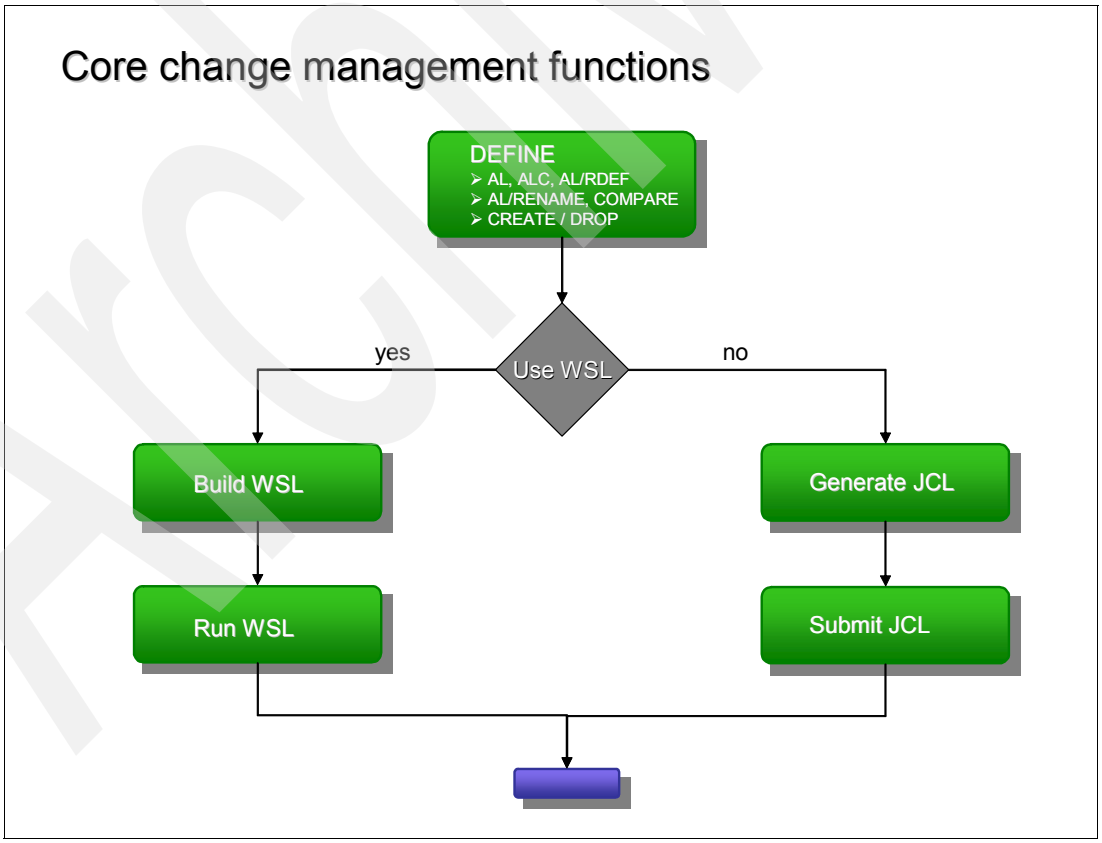

Figure 1-4 Core Change Management functions

Archived

Archived

Part 2

Getting started with Change Management

In this part of the book, we provide information related to the initial usage of DB2 Admin and Compare with the intent of setting up an environment in which changes must be managed.

This part consists of the following chapters:

- Chapter 2, “Customization” on page 15
- Chapter 3, “First steps with DB2 Admin” on page 45
- Chapter 4, “Working with the Change Management functions” on page 111

© Copyright IBM Corp. 2007. All rights reserved.

Archived

2

Customization

In this chapter we describe the tasks to be performed after the installation of the products — that is, after the libraries of DB2 Admin and DB2 Compare have been unloaded and the recommended maintenance has been applied to your system using the standard SMP/E installation process. Before you can start using DB2 Admin and DB2 Compare, you must evaluate and perform tasks that are related to product customization.

We describe the key customization tasks in some detail, we tell you why and how to set up a maintenance strategy for the products, and we provide information about the Change Management functions in terms of product tailoring and security aspects.

We cover the following topics:

- Installing DB2 Admin and DB2 Compare
- Maintenance strategy and deployment
- Tailoring for Change Management
- DB2 Security for Change Management

© Copyright IBM Corp. 2007. All rights reserved.

2.1 Installing DB2 Admin and DB2 Compare

After executing the SMP/E process, you are required to follow the normal installation tasks. These are described in detail in Chapter 2 of the *DB2 Administration Tool for z/OS User's Guide and Reference Version 7 Release 2,* SC18-9552.

You might also want to review Chapter 2 of the *DB2 Object Comparison Tool for z/OS User's Guide Version 7 Release 2,* SC18-9554; however, the whole installation tasks have to be done within DB2 Admin.

This section does not replace the descriptions provided in the *User's Guides*. The intent here is to give you additional advice in terms of the installation tasks where appropriate. It is not our intention to just repeat the installation tasks, nor to discuss DB2 version specific installation tasks.

2.1.1 z/OS related customization

There are several modules shipped in a library named *prefix*.SADBLINK. We strongly recommend that you authorize the library via Authorized Program Facility (APF) and put it into your linklist concatenation.

As the TSO service facility has to invoke ADB2ATH and ADB2UTIL as authorized programs, make sure that these two modules are registered for AUTHPGM and AUTHTSF in your SYS1.PARMLIB(IKJTSOxx). See Figure 2-1.

```
EDIT       SYS1.PARMLIB(IKJTS000) - 01.34          Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ****************************** Top of Data ******************************
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -   106 Line(s) not Displayed
000103 AUTHPGM NAMES(         /* AUTHORIZED PROGRAMS      */      +
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 42 Line(s) not Displayed
000146    ADB2ATH             /* CALLS DSN30ATH           */      +
000147    ADB2UTIL            /* CALLS DSNUTILB           */      +
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  1 Line(s) not Displayed
000149           )            /*                          */
000150                        /*                          */
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  5 Line(s) not Displayed
000156 AUTHTSF NAMES(         /* PROGRAMS TO BE AUTHORIZED */  +
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 22 Line(s) not Displayed
000179    ADB2ATH             /* CALLS DSN30ATH            */  +
000180    ADB2UTIL            /* CALLS DSNUTILB            */  +
000181           )            /*                          */
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 47 Line(s) not Displayed
****** **************************** Bottom of Data ****************************
```

Figure 2-1 The definition of ADB2ATH and ADB2UTIL in your SYS1.PARMLIB

There are also some new modules available with DB2 Admin and DB2 Compare V7.2: ADBMSGT, IADBMSGT, UADBMSGT, and ADB5000. These new modules use your Language Environment® and assist in translating messages to other languages. They do not have to be APF authorized.

ADB2ATH is a module that is used if you use your own DB2 security exit type. Refer also to Chapter 2, Step 8b of *DB2 Administration Tool for z/OS User's Guide and Reference Version 7 Release 2,* SC18-9552 if you use your own security exit.

Many changes do not just involve DDL, but also DB2 utilities. The utilities are invoked from a Work Statement List (WSL). ADB2UTIL is a program which is used when you run DB2 utilities from a WSL.

2.1.2 Sizing the objects for Change Management

Change Management is a new feature made available with DB2 Admin and DB2 Compare V7.2. *DB2 Administration Tool for z/OS User's Guide and Reference Version 7 Release 2,* SC18-9552, tells you to create all the Change Management related objects with *prefix*.SADBSAMP(ADBCHANG).

APAR PK36230 has been recently introduced to add a column to the ADBCHG table. The column name is RUNSQLID. You have to run *prefix*.SADBSAMP(ADBCHU01) after running *prefix*.SADBSAMP(ADBCHANG).

Defining the size of these new objects is the first task when you start enabling Change Management. Note that within the Change Management database there are entries to be stored called *versions*. Versions are basically definition instances of DB2 objects, saved as CM internal DDL statements in either DB2 tables or flat files. Their size is difficult to predict and it can quickly increase depending on the change activity. The corresponding Change Management Tables can grow quickly and currently there is no automatic way of getting rid of old entries.

Do not underestimate the entries in your Change Management database. Unless you are installing Change Management for your own basic testing purposes, we recommend not to use the defaults given in *prefix*.SADBSAMP(ADBCHANG).

We provide in Table 2-1 the starting size for your Change Management table spaces. You can keep the index quantities as 720 for PRIQTY and SECQTY except for the indexes ADBCVERLINESX1 and ADBCVERLINESX2, which can grow similarly to table ADBCVERLINES. Consider a starting value of 7200 for both PRIQTY and SECQTY for ADBCVERLINESX1 and ADBCVERLINESX2.

You can also consider the utilization of the sliding allocation function provided with DB2 for z/OS Version 8 by specifying the DSNZPARM parameter MGEXTSZ YES (with a global scope), option 7 on install panel DSNTIP7. See *Disk Storage Access with DB2 for z/OS*, REDP-4187.

Table 2-1 Table space allocation

Table space	PRIQTY in KB	SECQTY in KB
ADBSCHG	7200	7200
ADBSCHGS	72000	72000
ADBSCPRQ	720	720
ADBSCMSK	720	720
ADBSCIGN	720	720
ADBSCVER	7200	7200
ADBSCVLN	72000	72000

Table space	PRIQTY in KB	SECQTY in KB
ADBSCVSC	720	720
ADBSCID	720	720

Note: Besides object sizing, note that there are GRANT statements at the end of the Change Management DDL definition given in *prefix*.SADBSAMP(ADBCHANG). There are also GRANT statements given at the end of *prefix*.SADBSAMP(ADBBIND). It might be necessary to modify or delete the appended GRANT statements depending on how DB2 security has been established in your shop.

2.1.3 Creator for Change Management tables

Note that the definition of *owner* of the Change Management tables is important for the activation of Change Management. The owner is ADB by default. You can set the table owner to a value of your choice before submitting *prefix*.SADBSAMP(ADBCHANG).

Note: The job *prefix*.SADBSAMP(ADBCHANG) uses 'SET CURRENT SQLID' to set implicitly the qualifier of your tables. As this syntax also changes the SQL authorization ID, make sure that the value you set for 'SET CURRENT SQLID' has the appropriate DB2 authorization to create the Change Management objects.

If you do not want to proceed that way, you have to add the owner manually for Change Management tables and indexes.

We show how to activate Change Management with the appropriate owner at 2.1.7, "Enabling Change Management in your ISPF table". We mention them here, before you do the DB2 Admin BINDs, as they have to run with the created objects of Change Management.

2.1.4 DB2 BINDs

BINDs have to be run just for DB2 Admin, not for DB2 Compare. There is currently only a limited versioning available for your DB2 Admin packages. Basically there are no markers available to tell you the maintenance level of a package.

Therefore be careful when you do the BINDs with the same collection ID and another DBRM library. You might overwrite your existing packages.

You should adapt the *owner* of your BIND package statements for those packages that belong to change management. They are labeled as shown in Figure 2-2.

```
BIND PACKAGE(ADBL) -
     MEMBER(ADBCRCM) -
     QUALIFIER(ADB)      /*  THE QUALIFIER SHOULD MATCH THE OWNER */ -
     VALIDATE(BIND)      /*  OF THE CHANGE MANAGEMENT TABLES.     */ -
     ISOLATION(CS) -
     RELEASE(C) -
     ENCODING(EBCDIC)  -
     EXPLAIN(NO)
```

Figure 2-2 Adapt the owner of your Change Management tables

Refer to 2.2, “Maintenance strategy and deployment” on page 31 to see how you can best set up maintenance.

2.1.5 Running the customization EXEC

Running the customization exec *prefix*.SADBEXEC(ADB2CUST) is explained in Chapter 2 of *DB2 Administration Tool for z/OS User’s Guide and Reference Version 7 Release 2*, Step 8. In this section we describe why you have to be careful about the impact of specifying general parameters for DB2 Admin.

Let us start with invoking ADB2CUST by typing `EX 'prefix.SADBEXEC(ADB2CUST)'` from the TSO command prompt option **6**. See Figure 2-3 for the initial customization panel.

```
DB2 Admin ------------ DB2 Administration - Customization -------------- 20:16
Option ===> 1


Options:
   1 - General parameters for DB2 Admin
   2 - DB2 subsystem parameters for DB2 Admin
   3 - Copy general parameters and DB2 subsystem parameters

Library to be used for DB2 Admin customization ISPF tables:
  ISPF table library   ==> 'ADB.V7R2M0.SADBTLIB'
```

Figure 2-3 ADB2CUST Main Menu

We choose option **1** - General parameters for DB2 Admin.

The general parameters are shown in Figure 2-4.

Archived

```
DB2 Admin ------------- Customization - General Parameters ------------- 20:25
Option ===>

General parameters for DB2 Admin:

 Press ENTER to save parameters, or END to leave without saving parameters.

 DB2 security exit type               ==> STD      (STD,SAMPLE,AUTH,NOCALL,OWN)
 Copy version control table qualifier ==>          (table owner ID)
 System identification method         ==>          (JESID,SMFID,SYSNAME,NONE)
 Unit name for TSO work data sets     ==> SYSDA
 Unit name for batch data sets        ==> SYSDA
 Installation name                    ==>
 Node Name                            ==>
 JES3 system                          ==> NO       (Yes/No)
 Unicode translation technique        ==> ER       ('ER      ', etc)
 ISPF LLib1(not lnklst'd) ==> ISP.SISPLOAD
 ISPF LLib2(not lnklst'd) ==>
 ISPF Message Library     ==> ISP.SISPMENU
 ISPF Table Library       ==> ISP.SISPTENU
 DB2 Admin APF Library    ==>

Current SYSAFF information for this MVS system:
 JES ID:          NONE (found via JESID method)
 SMF ID:          SC63 (found via SMFID method)
 MVS system name: SC63 (found via SYSNAME method)
```

Figure 2-4 ADB2CUST General parameters

Make sure that your ISPF load, message, and table libraries (in bold in Figure 2-4) show up according to your naming standards. If not, you have to enter them appropriately. If you leave them blank, the skeleton JCL is not translated and JCL errors can occur later on.

For example, the *analyze process* (this process is discussed later) uses the skeleton *prefix*.SADBSLIB(ADB2SPFB) which contains JCL naming like SYS1.SISPMENU. As mentioned, if you do not fill out the ISPF environment, JCL problems might occur later in your Change Management jobs.

2.1.6 Invoking DB2 Admin

To make the product and ISPF libraries available to your ISPF session, you can use the *ADBL* CLIST which comes with DB2 Admin and change it to your requirements. You can find the ADBL CLIST in *prefix*.SADBCLST.

The only changes we made were to define the prefixes for our product libraries by changing the *PROD, PRODADD, LIBPRE,* and *LIBAPRE* values as shown in Figure 2-5.

- PROD overrides the high-level qualifier for all DB2 Admin product libraries, or you can edit the ADBL CLIST and specify the high-level qualifier in the PROD parameter.
- PRODADD is the high-level qualifier for additional product libraries to allocate in front of PROD(). The default is none.

- LIBPRE is the prefix for DB2 Admin libraries. This prefix designates the first set of characters (up to four) in the final qualifier of the DB2 Admin libraries. The default is SADB.
- LIBAPRE is the prefix for PRODADD() libraries. The default is none.

In alternative, or additionally to using PROD, PRODADD, and USERADD, you can use LIST and LISTPRE.

- LIST is the high-level qualifiers of additional libraries to allocate before PROD(), PRODADD(), and USERADD().
- LISTPRE is the list of prefixes for LIST() libraries.

Our naming standards for the DB2 Admin libraries are *ADB.V7R2M0*, and for the DB2 Compare libraries we use *GOC.V7R2M0*. Your library names can be different.

```
EDIT       ADB.V7R2M0.SADBCLST(ADBL) - 01.07              Columns 00001 00072
Command ===>                                                 Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 PROC 0 SHOW DEBUG SYSTEM() PLAN() PANEL() PGM() +
000002   NEWAPPL() DUMP() LISTPARM +
000003   DB2LLIB() CMOWN() +
000004   PROD('ADB.V7R2M0') USER VB +
000005   PRODADD('GOC.V7R2M0') LIBPRE(SADB) LIBAPRE(SGOC) +
000006   USERADD() USERPRE() +
000007   LIST() LISTPRE() +
000008   INSTALL() SECEXIT() VIO() DASD() JES() DMT LIBDEF(YES)
```

Figure 2-5 A sample of the ADBL Clist

Figure 2-5 shows what your ADBL Clist should look like. There are plenty of additional modifications possible, which are described in Chapter 2, Step 6 of *DB2 Administration Tool for z/OS User's Guide and Reference Version 7 Release 2.*

The most important parameters are bolded in Figure 2-5. We recommend that you set LIBDEF to YES unless you do not care about allocating your product libraries.

Note: If you do not specify a plan name, the default plan *ADB* is taken as default. If you want to use two or more different versions of DB2 Admin on the same DB2 subsystem, you have to use another plan name for the other versions. In this case, modify PLAN() to your requirements.

You do not have to update the ADBL Clist in your product libraries directly. In Figure 2-6, you can see how to add the *PROD, PRODADD, LIBPRE,* and *LIBAPRE* parameters to your ADBL execution *as an argument*. By doing so, these parameters overwrite your ADBL settings when you invoke DB2 Admin. Your *prefix*.SADBCLST(ADBL) remains unchanged.

In summary, now you have the choice to either modify *prefix*.SADBCLST(ADBL) or keep it as delivered and invoke the product by the small REXX exec shown in Figure 2-6.

```
EDIT       SYS1.OS390.CLIST(A72) - 01.03                 Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** **************************** Top of Data ******************************
000100 /** REXX ***************************************************************/
000200 /* DB2 ADMIN AND COMPARE STARTER EXEC                                  */
000300 /**********************************************************************/
000400    PARSE UPPER ARG PARM
000500
000600    IF POS(' PROD(',' 'PARM)=0 THEN
000700      PARM = PARM 'PROD(ADB.V7R2M0) PRODADD(GOC.V7R2M0)' ,
000800                  'LIBPRE(SADB) LIBAPRE(SGOC)'
000900    ADDRESS 'ISPEXEC' 'SELECT MODE(FSCR) NEWAPPL(ADB)',
001000     'CMD(EX ''ADB.V7R2M0.SADBCLST(ADBL)'' '''PARM''')'
001100
001200    EXIT RC
****** **************************** Bottom of Data ***************************
```

Figure 2-6 A sample REXX™ to invoke ADBL including an argument

Note: Take notice of the support given by the parameter NEWAPPL shown in Figure 2-6. When using the REXX exec here, this parameter is always set to ADB. As DB2 Admin stores default settings (such as a job card) in your standard ISPF profile library, those DB2 Admin defaults are always taken from the same ISPF profile member.

This guarantees that, whenever you invoke the DB2 Admin REXX exec, you are always using your defaults. It does not matter if you invoke DB2 Admin from the ISPF Main Menu or from another panel. You do not have to use ADB, you can just adapt the prefix to your requirements.

Archived

If all your settings have been done correctly, the general DB2 Admin Main Menu is presented. See Figure 2-7.

```
 DB2 Admin -------------- DB2 Administration Menu 7.2.0 ------------------
11:27
 Option ===>

    1 - DB2 system catalog                                DB2 System: DB8A
    2 - Execute SQL statements                            DB2 SQL ID: PAOLOR1
    3 - DB2 performance queries                           Userid    : PAOLOR1
    4 - Change current SQL ID                             DB2 Rel   : 815
    5 - Utility generation using LISTDEFs and TEMPLATEs
    P - Change DB2 Admin parameters
   DD - Distributed DB2 systems
    E - Explain
    Z - DB2 system administration
   SM - Space management functions
    W - Manage work statement lists
    X - Exit DB2 Admin



 Interface to other DB2 products and offerings:
    D  - DB2I
```

Figure 2-7 DB2 Admin Main Menu

2.1.7 Enabling Change Management in your ISPF table

In this section we show how to set up the ISPF table to enable Change Management within DB2 Admin.

After invoking ADB2CUST by typing `EX 'prefix.SADBEXEC(ADB2CUST)'` from the TSO command prompt option **6**, you choose option **2** as shown in Figure 2-8. Make sure that the entry for the ISPF table library corresponds to the one you use in your DB2 Admin and DB2 Compare environment.

```
DB2 Admin ------------ DB2 Administration - Customization -------------- 13:44
Option ===> 2


Options:
   1 - General parameters for DB2 Admin
   2 - DB2 subsystem parameters for DB2 Admin
   3 - Copy general parameters and DB2 subsystem parameters

Library to be used for DB2 Admin customization ISPF tables:
  ISPF table library   ==> 'ADB.V7R2M0.SADBTLIB'
```

Figure 2-8 ADB2CUST Main Menu

Archived

Choose option **2** to edit your ISPF table entries in the panel of Figure 2-9.

```
DB2 Admin --------- Customization - DB2 Subsystem Parameters ----------- 13:49
Option ===> 2


Options:
   1 - Merge list of active DB2 subsystem(s) into ISPF table
   2 - Edit ISPF table

Currently active DB2 subsystem(s) on this MVS system:
  DB2 subsystem(s):    DB8A DB9B D8F1

Data sets to be used for DB2 Admin customization:
  ISPF table library:  'ADB.V7R2M0.SADBTLIB'
```

Figure 2-9 Choosing the ISPF table

After invoking the ISPF table, you get a list of tags that help to easily maintain your DB2 subsystem entries. DB2 Admin V7.2, introduces a new tag named **:cmown**. You can see this new tag in Figure 2-10.

```
EDIT       SYS07058.T150007.RA000.PAOLOR1.R0137646         Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001
000002 *
000003 * Tags recognized by ADB2CUST EXEC:
000004 * ---------------------------------------------------------------------
000005 * :nick.     Name of DB2 subsystem or DB2 group being described
000006 * :desc.     Text to be displayed for this DB2 subsystem or DB2 group
000007 * :secexit.  DB2 security exit type (STD,SAMPLE,AUTH,OWN,NOCALL)
000008 * :cvtown.   Copy version table owner
000009 * :stcnam.   Started task name of this DB2 subsystem
000010 * :grpnam.   DB2 group name of this DB2 subsystem or DB2 group
000011 * :jclass.   Job class to be used for batch DB2 utility jobs
000012 * :jsysaff.  SYSAFF to be used for batch DB2 utility jobs
000013 * :instparm. Installation name
000014 * :utilpre.  Utility data set prefix (USERID,OWNER,CREATEDBY,name)
000015 * :ssid.     Subsystem name of the remote subsystem (blank if local)
000016 * :nodenam.  Node name of the remote subsystem (blank if local)
000017 * :locnam.   Location name of the remote subsystem (blank if local)
000018 * :authsw.   Authorization Switching enabled (YES,NO)
000019 * :newappl.  ISPF application ID
000020 * :prompt.   Default value for Prompt Options (YES,NO)
000021 * :resetopt. Default value for Reset to Default at Startup (YES,NO)
000022 * :nstuproc. Number of job steps in the DSNUPROC procedure
000023 * :cmown.    Owner (Qualifier) of Change Management database objects
000024 * :hpu.      High Performance Unload (HPU) enabled (YES,NO)
000025 * :hpullib.  High Performance Unload (HPU) load library
000026 *
000027 * :bl2llib.  DB2 load library
```

Figure 2-10 Tag summary of your ISPF table

Press Enter to eliminate the truncation warning of variable length data.

The new tag **:cmown** reflects your creator from the tables of the Change Management database. Whenever Change Management is used, it refers to the Change Management tables using the content of :cmown as your Change Management table owner. Figure 2-11 shows our ISPF table.

```
EDIT       SYS07058.T180723.RA000.PAOLOR1.R0137742        Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ***************************** Top of Data ******************************
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 48 Line(s) not Displayed
000049 :nick.DB8A  :desc.DB2 Version 8 Testsystem for CM Residency
000050             :jclass.A
000051             :authsw.NO
000052             :cmown.A72
000053             :hpu.NO
000054             :bl2llib.'DB8A8.SDSNLOAD'
000055                      'DB8A8.SDSNEXIT'
000056             :bl2rlib.'DB8AU.RUNLIB.LOAD'
000057             :bl2mlib.'DB2G7.SDSNSPFM'
000058             :bl2plib.'DB2G7.SDSNSPFP'
000059                      'DB2G7.SDSNPFPE'
000060             :bl2clib.'DB2G7.SDSNCLST'
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 28 Line(s) not Displayed
****** **************************** Bottom of Data ****************************
```

Figure 2-11 The tag :cmown has been set to the value 'A72'

Archived

Figure 2-12 shows our corresponding Change Management tables.

```
DB2 Admin --------------- DB8A Tables, Views, and Aliases -------- Row 1 of 13
Command ===>                                                  Scroll ===> CSR

Commands: GRANT  MIG
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
 V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
 ? - Show all line commands

Sel   Name                Owner    T DB Name  TS Name    Cols        Rows Checks
      *                   *        * *        *            *           *      *
----- ------------------ -------- - -------- -------- ------ ----------- ------
      ADBCHG              A72      T A72DCHG  ADBSCHG      22          -1      0
      ADBCHGS             A72      T A72DCHG  ADBSCHGS      9          -1      0
      ADBCHGSR            A72      T A72DCHG  ADBSCHGS      6          -1      0
      ADBCPREREQ          A72      T A72DCHG  ADBSCPRQ      7          -1      0
      ADBCMASK            A72      T A72DCHG  ADBSCMSK      8          -1      0
      ADBCMASKS           A72      T A72DCHG  ADBSCMSK      5          -1      0
      ADBCIGNORE          A72      T A72DCHG  ADBSCIGN      8          -1      0
      ADBCIGNORES         A72      T A72DCHG  ADBSCIGN      3          -1      0
      ADBCVERSION         A72      T A72DCHG  ADBSCVER     33          -1      0
      ADBCVERLINES        A72      T A72DCHG  ADBSCVLN     11          -1      0
      ADBCVERSCOPE        A72      T A72DCHG  ADBSCVSC      8          -1      0
      ADBCVERSCOPES       A72      T A72DCHG  ADBSCVSC      5          -1      0
      ADBCID              A72      T A72DCHG  ADBSCID       2          -1      0
****************************** END OF DB2 DATA ******************************
```

Figure 2-12 Change Management tables

Once the **:cmown** tag has been defined for your subsystem(s), Figure 2-13 shows the newly added option **CM** in your DB2 Admin Main Menu. **CM** is the abbreviation for Change Management.

```
DB2 Admin -------------- DB2 Administration Menu 7.2.0 ------------------ 12:52
Option ===>

   1 - DB2 system catalog                                DB2 System: DB8A
   2 - Execute SQL statements                            DB2 SQL ID: PAOLOR1
   3 - DB2 performance queries                           Userid    : PAOLOR1
   4 - Change current SQL ID                             DB2 Rel   : 815
   5 - Utility generation using LISTDEFs and TEMPLATEs
   P - Change DB2 Admin parameters
  DD - Distributed DB2 systems
   E - Explain
   Z - DB2 system administration
  SM - Space management functions
   W - Manage work statement lists
   X - Exit DB2 Admin

  CM - Change management
```

Figure 2-13 DB2 Admin Main Menu with Change Management enabled

2.1.8 Disabling Change Management

You have seen how you can enable Change Management for your subsystems with the new tag **:cmown**. Change Management is enabled just for those subsystems where you put this tag into your ISPF table. There is also a possibility to disable Change Management overall. This is not done within the ISPF table, but within your ADBL Clist. You can find the same keyword name CMOWN in the ADBL Clist.

After invoking DB2 Admin, you can modify CMOWN with a dash '-' as shown in Figure 2-14. CMOWN(-) disables Change Management for all the subsystems.

```
EDIT       ADB.V7R2M0.SADBCLST(ADBL) - 01.14              Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 PROC 0 SHOW DEBUG SYSTEM() PLAN(ADB) PANEL() PGM() +
000002   NEWAPPL() DUMP() LISTPARM +
000003   DB2LLIB() CMOWN(-) +
000004   PROD() USER VB +
000005   PRODADD() LIBPRE() LIBAPRE() +
000006   USERADD() USERPRE() +
000007   LIST() LISTPRE() +
000008   INSTALL() SECEXIT() VIO() DASD() JES() DMT LIBDEF(YES)
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -   899 Line(s) not Displayed
****** **************************** Bottom of Data ****************************
```

Figure 2-14 Disabling Change Management

2.1.9 Specifying DB2 Compare on the DB2 Admin Main Menu

You can invoke DB2 Compare from the DB2 Admin Main Menu by doing some customization inside your ISPF table. Jump to your ISPF table as we have shown at 2.1.7, "Enabling Change Management in your ISPF table" on page 23. Refer to the description in Figure 2-9 on page 24. You can see the tag summary (refer to Figure 2-10 on page 24) and then scroll down to see your subsystem entries.

Assuming that you just want to add DB2 Compare as an interface, you add into your ISPF table the bolded lines shown in Figure 2-15.

Archived

```
EDIT       SYS07059.T115923.RA000.PAOLOR1.R0138327         Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
****** ***************************** Top of Data ******************************
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 50 Line(s) not Displayed
000049 :nick.DB8A  :desc.DB2 Version 8 Testsystem for CM Residency
000050             :jclass.A
000051             :authsw.NO
000052             :resetopt.YES
000053             :cmown.A72
000054             :hpu.NO
000055             :bl2llib.'DB8A8.SDSNEXIT'
000056                      'DB8A8.SDSNLOAD'
000057             :bl2rlib.'DB8AU.RUNLIB.LOAD'
000058             :bl2mlib.'DB2G7.SDSNSPFM'
000059             :bl2plib.'DB2G7.SDSNSPFP'
000060                      'DB2G7.SDSNPFPE'
000061             :bl2clib.'DB2G7.SDSNCLST'
000062             :aopt.C
000063             :adescr.DB2 Object Comparison Tool
000064             :apan.GOCMENU
000065             :bopt.I
000066             :bdescr.DB2I
000067             :bispf.SELECT CMD(%DSNECPRI SSID(&DB2SYS)) NEWAPPL(DSNE) PAS
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 19 Line(s) not Displayed
****** **************************** Bottom of Data ****************************
```

Figure 2-15 Adding DB2 Compare to your ISPF table

After saving the ISPF table, your DB2 Admin Main Menu looks as shown in Figure 2-16.

```
DB2 Admin -------------- DB2 Administration Menu 7.2.0 ------------------ 18:44
Option ===>

   1 - DB2 system catalog                                DB2 System: DB8A
   2 - Execute SQL statements                            DB2 SQL ID: PAOLOR1
   3 - DB2 performance queries                           Userid    : PAOLOR1
   4 - Change current SQL ID                             DB2 Rel   : 815
   5 - Utility generation using LISTDEFs and TEMPLATEs
   P - Change DB2 Admin parameters
  DD - Distributed DB2 systems
   E - Explain
   Z - DB2 system administration
  SM - Space management functions
   W - Manage work statement lists
   X - Exit DB2 Admin
  CC - DB2 catalog copy version maintenance
  CM - Change management

Interface to other DB2 products and offerings:
   C  DB2 Object Comparison Tool
   I  DB2I
```

Figure 2-16 DB2 Admin Main Menu

We just have shown you how to prepare the invocation of DB2 Compare inside your ISPF table. In order to run DB2 Compare from the DB2 Admin Main Menu, the product libraries from DB2 Compare have to be concatenated as well. You can use the variables *PRODADD* and *LIBAPRE* in your ADBL Clist. Refer to 2.1.6, “Invoking DB2 Admin” on page 20 to learn about how to use these variables.

2.1.10 Additional information for your ISPF table

Notice that, if you add more options to your ISPF table, the first letter of :a, :b, ... is responsible for the sequence that is later shown in your DB2 Admin Main Menu. Figure 2-15 on page 28 shows that we are adding two options to our environment. Option `C` is for DB2 Compare and option `I` is for SPUFI.

It is worth mentioning that if your DB2 exit and load library are not concatenated to your linklist you have to list them within the `:b12llib` tag of your ISPF table. Make sure that you specify the *prefix*.SDSNEXIT library first and then the *prefix*.SDSNLOAD library. You can see this in Figure 2-15 on page 28, too.

2.1.11 Running two different versions in parallel (co-existence)

You might require two versions of DB2 Admin and DB2 Compare installed on your system. Let us assume that you want to restrict the new release DB2 Admin and DB2 Compare V7.2 to a limited number of users in order to do some testing scenarios. On the other hand, you want to keep the already installed release because the product itself and all the processes have been established in your environment.

When running two different versions in parallel, there are some aspects you have to consider.

Linklist modules

As we have shown in 2.1.1, “z/OS related customization” on page 16, there are several modules which have to be put into the linklist concatenation of your system. As a reminder, these are the modules from *prefix*.SADBLINK. When running two different versions in parallel, make sure that you always put the modules from the highest version of *prefix*.SADBLINK into your linklist concatenation.

DB2 BINDs

Be careful not to overwrite the existing packages of DB2 Admin. Note that DB2 Compare does not have its own DB2 packages. Especially when running two different versions of DB2 Admin and DB2 Compare in parallel, you must define unique names for the following items:

- Collection name of the DB2 Admin packages
- Plan name of the DB2 Admin plans
- PKLIST entries of the DB2 Admin plans

You can find the provided BIND JCL in *prefix*.SADBSAMP(ADBBIND). Make sure to provide unique names for collections, plans, and PKLIST entries before submitting the bind job.

2.1.12 Quick start customization summary

If this is your first time with DB2 Admin and DB2 Compare and you want to get a basic understanding of DB2 Admin and DB2 Compare, as well as using Change Management, you do not have to perform all of the installation steps described in *DB2 Administration Tool for z/OS User’s Guide and Reference Version 7 Release 2,* SC18-9552.

We provide Table 2-2 to let you know which steps are absolutely necessary and which ones you can do later. Again, after doing the mandatory marked installation tasks, you can start to get used to the tools and Change Management.

Table 2-2 Summary of installation steps

Step	Task keywords	Mandatory for quick start	Not mandatory for quick start
1	Checkpoint table	X	
2	TEMP Database	X	
3	CM database objects	X	
4	DB2 Binds	X	
5	FB to VB		X
6	Prepare ADBL	X	
7	Early verify		X
8a	Run ADB2CUST	X	
8b	Customize ISPF table	X	
9a	Adapt Main Menu		X
9b	Adapt DB2 Selection Menu		X
9c	Tailor 9a and 9b by usermods		X
10	ADB2UCUS customization		X
11	Launchpad		X
12	MVS tasks	X	
13	Catalog Copy		X
14	Catalog views		X
15	Catalog access		X
16	Stored Procedure CC/390		X
17	Catalog indexes		X
18	Adapt Zparm for DSNWZP		X
19	RUNSTATS DB2 catalog		X
20	Line commands		X
21	Enable Auth switching		X
22	Verify additional functionality	X	
23	DB2 Admin for all users available		X
24	DB2 Release upgrade		X
25	Utility templates		X

Archived

2.2 Maintenance strategy and deployment

In this section we show you how to implement a maintenance strategy for DB2 Admin and DB2 Compare. Our maintenance strategy is just an example of many possible ones you can use, and you might have to adapt it to your shop's standards. The one we describe has been established in a customer's environment.

As naming standards in your environment most likely vary, we do not provide generated JCL in this section. We give you an overview of the tasks that have to be implemented to follow an automated procedure when applying maintenance to the product. You can follow this procedure for a single PTF, a bundle of PTFs, and a new software level of DB2 Admin and DB2 Compare.

Note: Especially when you install a new software level, make sure that you apply all the available maintenance to the products. Sometimes the shipped PTFs are not just for fixing some pieces of code, but also are a vehicle for new or enhanced product functions that you might want to benefit from.

2.2.1 Building a new SMP/E package

You basically have the typical SMP/E zones available in your SMP/E environment: global zone (GLOBAL), target zone (TARG) and distribution zone (DLIB). Make sure that you have already received and applied the available maintenance into TARG.

Now take TARG and DLIB and clone them with the SMP/E command ZONECOPY to a new name. Let us call this new name *newTARG* for your new target zone and *newDLIB* for your new distribution zone.

Note that your zones TARG and DLIB are combined with data sets. If you clone those zones to *newTARG* and *newDLIB,* you have to clone the corresponding data sets as well. Make sure that they match, i.e. *prefix.targ.*SADBSLIB must be cloned to *prefix.newtarg.*SADBSLIB.

After doing the ZONECOPY, you can then perform an ADRDSSU DUMP DSN (...) command to dump all the data sets from *newTARG* to a flat file. You can use this flat file to transfer it to the environment of your choice where you plan to deploy your new software or maintenance level.

Archived

We illustrate the workflow in Figure 2-17.

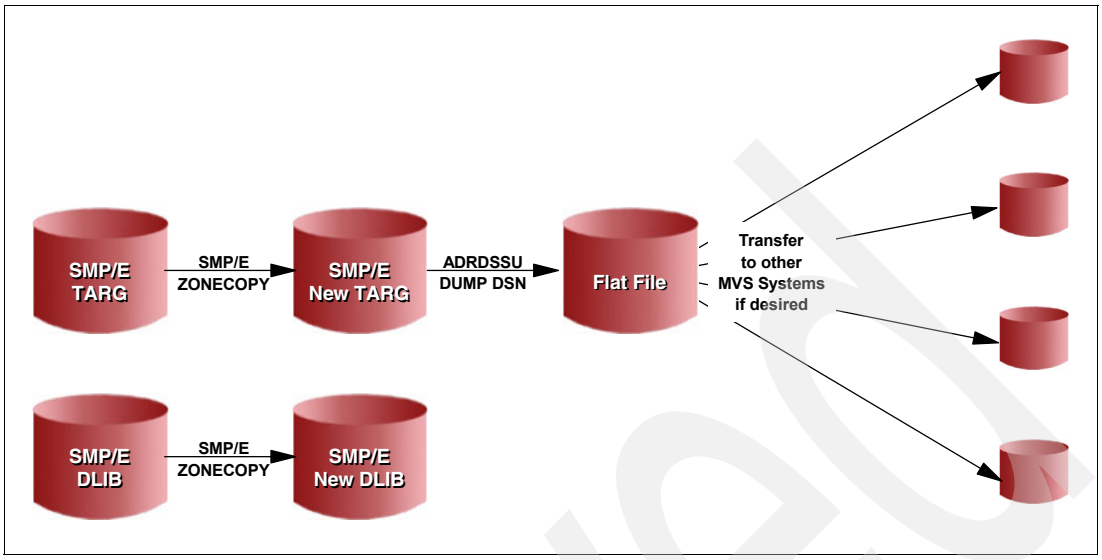

Figure 2-17 Maintenance strategy - Initial approach

2.2.2 Restoring the flat file

You then use this flat file to restore all the dumped data sets to a new runtime naming convention of your DB2 Admin and DB2 Compare environment. You should mark all the restored new data sets with a new *unique qualifier*, such as P1. See Example 2-1 for what these restored data sets might look like.

Example 2-1 A runtime environment for DB2 Admin and DB2 Compare

```
ADB.RUNTIME.P1.SADBBASE
ADB.RUNTIME.P1.SADBCLST
ADB.RUNTIME.P1.SADBDBRM
ADB.RUNTIME.P1.SADBEXEC
ADB.RUNTIME.P1.SADBLINK
ADB.RUNTIME.P1.SADBLLIB
ADB.RUNTIME.P1.SADBMLIB
ADB.RUNTIME.P1.SADBPLIB
ADB.RUNTIME.P1.SADBSAMP
ADB.RUNTIME.P1.SADBSLIB
ADB.RUNTIME.P1.SADBTLIB
GOC.RUNTIME.P1.SGOCBASE
GOC.RUNTIME.P1.SGOCCLST
GOC.RUNTIME.P1.SGOCEXEC
GOC.RUNTIME.P1.SGOCLLIB
GOC.RUNTIME.P1.SGOCMLIB
GOC.RUNTIME.P1.SGOCPLIB
GOC.RUNTIME.P1.SGOCSAMP
GOC.RUNTIME.P1.SGOCSLIB
GOC.RUNTIME.P1.SGOCTLIB
```

Archived

2.2.3 Doing the BINDs

You now do your DB2 BINDs because your new runtime load library and the corresponding DB2 packages must work together. So you must use your new runtime environment ADB.RUNTIME.P1.SADBDBRM as DBRM input to your BIND PACKAGE statements. There are no BINDs necessary for DB2 Compare. Your unique qualifier P1 should be reflected in your collection as well. Do not use the provided collection ADBL. Instead, use ADBL_P1.

Do not forget to adapt your DB2 Admin plan names with these new collections. You only have to do this task once. If you plan to have a maximum of four runtime environments, your PKLIST entries should look like Figure 2-18.

```
VIEW       ADB.V7R2M0.SADBSAMP(ADBBIND) - 01.02           Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ***************************** Top of Data ******************************
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 84 Line(s) not Displayed
007500  BIND PACKAGE(ADBL_P1) -
007600       MEMBER(ADBMAIN) -
007700       VALIDATE(BIND) -
007800       ISOLATION(CS) -
007900       RELEASE(C) -
008000       ENCODING(EBCDIC)  -
008100       EXPLAIN(NO)
008200
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -   387 Line(s) not Displayed
030100  BIND PLAN(ADB27SPC) -
030200       PKLIST(ADBL_P1.*, ADBL_P2.*, ADBL_P3.*, ADBL_P4.*) -
030300       VALIDATE(BIND) -
030400       ISOLATION(CS) -
030500       ACQUIRE(USE) -
030600       ENCODING(EBCDIC)  -
030700       EXPLAIN(NO)
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  5 Line(s) not Displayed
****** **************************** Bottom of Data ****************************
```

Figure 2-18 BIND PACKAGE and BIND PLAN

2.2.4 Invoking DB2 Admin with the new maintenance level

Instead of invoking the Clist ADBL directly from your new runtime environment marked with P1, you should always call ADBL by an alias. This allows you to keep your invocation procedure of DB2 Admin always with the same name. You only have to change the name where the alias name points to. In our example, we want to create alias names for the whole runtime environment pointing to P1. Refer to Example 2-2. Notice the difference: P0 represents the alias, P1 represents the runtime data set name.

Example 2-2 Alias names referring to your runtime environment

```
ADB.RUNTIME.P0.SADBBASE is an alias and points to ADB.RUNTIME.P1.SADBBASE
ADB.RUNTIME.P0.SADBCLST is an alias and points to ADB.RUNTIME.P1.SADBCLST
ADB.RUNTIME.P0.SADBDBRM is an alias and points to ADB.RUNTIME.P1.SADBDBRM
ADB.RUNTIME.P0.SADBEXEC is an alias and points to ADB.RUNTIME.P1.SADBEXEC
ADB.RUNTIME.P0.SADBLINK is an alias and points to ADB.RUNTIME.P1.SADBLINK
ADB.RUNTIME.P0.SADBLLIB is an alias and points to ADB.RUNTIME.P1.SADBLLIB
ADB.RUNTIME.P0.SADBMLIB is an alias and points to ADB.RUNTIME.P1.SADBMLIB
```

```
ADB.RUNTIME.PO.SADBPLIB is an alias and points to ADB.RUNTIME.P1.SADBPLIB
ADB.RUNTIME.PO.SADBSAMP is an alias and points to ADB.RUNTIME.P1.SADBSAMP
ADB.RUNTIME.PO.SADBSLIB is an alias and points to ADB.RUNTIME.P1.SADBSLIB
ADB.RUNTIME.PO.SADBTLIB is an alias and points to ADB.RUNTIME.P1.SADBTLIB

GOC.RUNTIME.PO.SGOCBASE is an alias and points to GOC.RUNTIME.P1.SGOCBASE
GOC.RUNTIME.PO.SGOCCLST is an alias and points to GOC.RUNTIME.P1.SGOCCLST
GOC.RUNTIME.PO.SGOCEXEC is an alias and points to GOC.RUNTIME.P1.SGOCEXEC
GOC.RUNTIME.PO.SGOCLLIB is an alias and points to GOC.RUNTIME.P1.SGOCLLIB
GOC.RUNTIME.PO.SGOCMLIB is an alias and points to GOC.RUNTIME.P1.SGOCMLIB
GOC.RUNTIME.PO.SGOCPLIB is an alias and points to GOC.RUNTIME.P1.SGOCPLIB
GOC.RUNTIME.PO.SGOCSAMP is an alias and points to GOC.RUNTIME.P1.SGOCSAMP
GOC.RUNTIME.PO.SGOCSLIB is an alias and points to GOC.RUNTIME.P1.SGOCSLIB
GOC.RUNTIME.PO.SGOCTLIB is an alias and points to GOC.RUNTIME.P1.SGOCTLIB
```

Figure 2-19 shows how this looks using a diagram. Once your flat file is available in the target environment, the libraries have to be restored from the flat file with the unique qualifier P1 and your runtime naming convention. You then have to create an alias pointing to each runtime library.

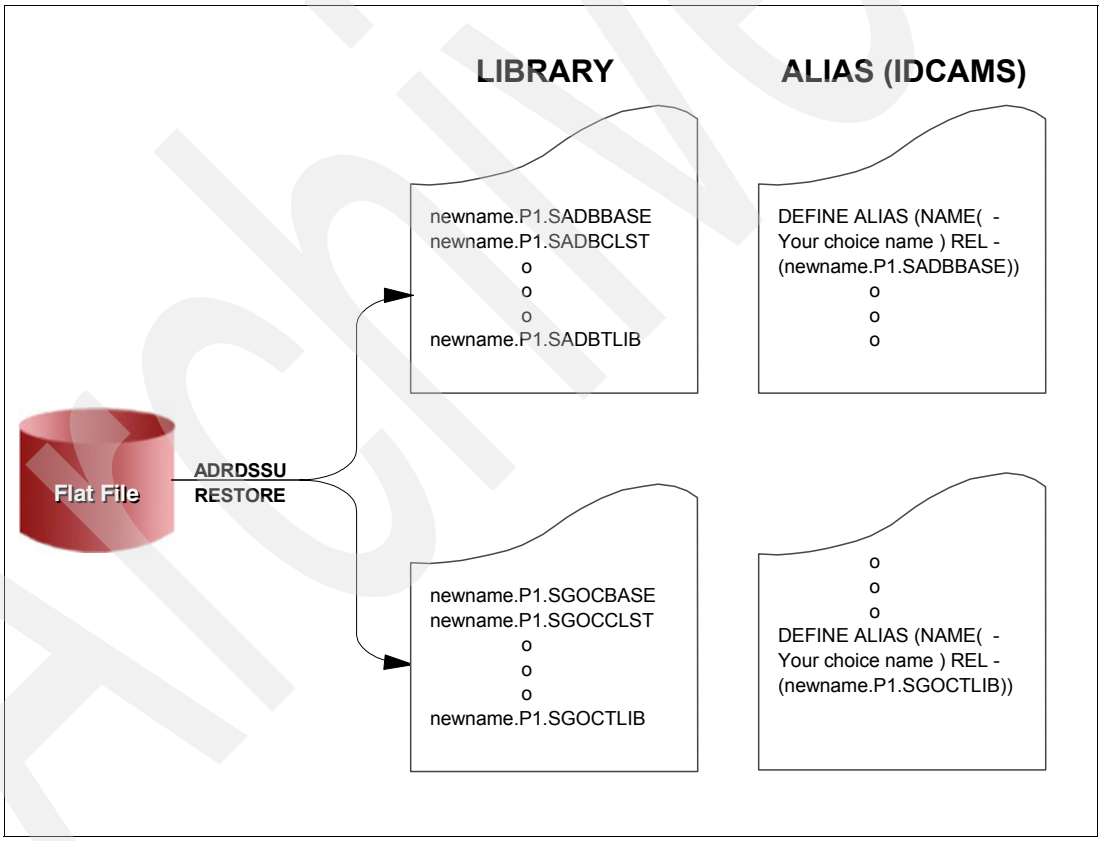

Figure 2-19 Restoring the files and creating aliases

Note: You basically can have another set of restored libraries with another unique qualifier 'P2' representing another level of maintenance or a new software level. Most likely it makes sense to use P1, P2, P3, and P4 and then starting again with P1.

2.2.5 Deploying your new maintenance level to another LPAR

You can easily transfer just the dumped flat file to another LPAR. You can then follow the same procedure as described, restoring to a new runtime environment, doing the BINDs and pointing your alias name to this new runtime environment.

2.2.6 Advantages of this maintenance strategy

Let us assume that all your alias names point to your P1 runtime environment. You prepared a new runtime environment P2 recently and you did already the necessary DB2 BINDs with the collection ADBL_P2. Now you do a 'switch' of your alias names, which means that you delete the old alias names and create them again but this time they point to your recently created runtime environment P2.

If necessary, you can easily switch back to your old runtime environment P1 just by pointing your alias names back to them. All the required DB2 packages ADBL_P1 are still around and not overwritten.

Another advantage is that you can anticipate your new maintenance level. This means you can do all the DB2 BINDs already before your new runtime environment becomes active. For instance, your currently active runtime environment is based on P1, including your packages ADBL_P1. When you prepare the new runtime environment P2, you can already do the package bind for ADBL_P2 without interfering the current active runtime environment P1.

2.3 Tailoring for Change Management

In this section we describe what you have to do when you start working with Change Management.

> **Note:** As a prerequisite to use Change Management, you must have installed DB2 Compare V7.2. Many generated jobs are based on functions of DB2 Compare V7.2. You should set up your invocation of DB2 Admin including the libraries for DB2 Compare as shown in 2.1.6, "Invoking DB2 Admin" on page 20. Additionally, make sure that you configured your ISPF table as described in 2.1.7, "Enabling Change Management in your ISPF table" on page 23.

2.3.1 Managing the SQL ID table

It is important that change management strategy be well planned. The most important factor is to ensure that changes to a set of objects are either all performed through Change Management or are all performed without Change Management.

There are two ways to set up which SQL IDs can use Change Management:

- Default Change Management level for SQL IDs
- Setting Change Management level for specific SQL IDs

The three choices of levels are as follows:

- REQUIRED:

  All changes must be registered in the Change Management database.

- OPTIONAL:

  Changes can be registered in the Change Management database. You are prompted as to whether to make the change through Change Management or not. OPTIONAL might be used when you are testing Change Management or when you can ensure that the SQL ID will register the changes to objects that are being managed under Change Management when prompted.

- NONE:

  No changes can be registered in the Change Management database.

Possible examples of Change Management strategies are as follows:

- When CM is to be used for only a single application, only a few SQL IDs are allowed to make changes, and these changes must be managed in CM:
  - Set the default Change Management level to NONE for most of the SQL IDs.
  - Use the Change Management ID table to set a few SQL IDs to REQUIRED for any SQL ID that is managing the application changes.
- When CM is to be used for all applications except for a few objects that are under design and development:
  - Set the default CM level to REQUIRED.
  - Use the CM ID table to set only the SQL IDs that are used to change the objects that are under design and development to NONE. Or if these SQL IDs will also be changing other objects not in design or development, then they can be set to OPTIONAL. The user must then specify whether to use CM upon each change.
- When CM is to be tested:
  - Set the default level to OPTIONAL.

We now discuss an example of how to update the Change Management SQL ID table.

If the CM default setting is optional or required, then all SQL IDs would be prompted to use CM or required to use CM, unless there is a specific entry in the table for a particular SQL ID that sets it to something other than the default setting.

Working with Change Management is allowed just for those SQL IDs which are registered in the ID table :cmown.ADBCID of the Change Management database. After invoking Change Management with the primary command CM from the DB2 Admin Main Menu, you choose option **5** to enter the ID table.

Figure 2-20 shows what the SQL ID table looks like.

```
DB2 Admin -------------- CM - Manage ID Table --------------------- Row 1 of 3
Command ===>                                                Scroll ===> CSR

Commands: SAVE
Line commands:  D - Delete  I - Insert

Change management default level setting : NONE


Sel SQL ID   Level    Oper
    *        *        *
--- -------- -------- ------
    PAOLOR1  OPTIONAL
    PAOLOR2  OPTIONAL
    PAOLOR3  OPTIONAL
****************************** END OF DB2 DATA ******************************
```

Figure 2-20 The SQL ID table

Basically, you start with no SQL ID registered in the ID table. You can insert a new SQL ID by typing 'I' anywhere in the first column named with 'Sel'. Then you see two question marks and you are asked to enter the SQL ID and a level. This is shown in Figure 2-21. You are not asked to update column 'Oper', this reflects the activity you are currently doing, in this case INSERT.

```
DB2 Admin -------------- CM - Manage ID Table --------------------- Row 1 of 4
Command ===>                                                Scroll ===> CSR

Commands: SAVE
Line commands:  D - Delete  I - Insert

Change management default level setting : NONE


Sel SQL ID   Level    Oper
    *        *        *
--- -------- -------- ------
    PAOLOR1  OPTIONAL
    PAOLOR2  OPTIONAL
*   PAOLOR3  OPTIONAL
    ?        ?        INSERT
****************************** END OF DB2 DATA ******************************
```

Figure 2-21 Inserting a new SQL ID

Here you can set one of the three mentioned levels for the SQL ID you are inserting. They are defined as follows:

- REQUIRED - Changes must be registered for the given SQL ID:

  Whatever changes you make to your DB2 structures, they are always registered in the Change Management Database according to the SQL ID you mentioned. The change is registered in any case in the Change Management Database.

Archived

- OPTIONAL - Changes can be registered for the given SQL ID:

  Whenever you plan to make a change to your DB2 structures, you are asked if you want to perform this change with Change Management or if you want to arrange this change without registering in the Change Management Database.

- NONE - Changes are not registered for the given SQL ID:

  Change Management is disabled for the mentioned SQL ID. Nothing is registered.

> **Note:** Whenever you update your ID table, enter either the SAVE primary command or point-and-shoot on SAVE to save your modifications.

2.3.2 Setting the default CM level

The default CM level is used to describe the default for all SQL IDs that are not listed in the SQLID table. You can override the default value for all SQL IDs by adding a definition to the table for a particular SQL ID. You can set the default level to NONE for those SQL IDs that must not use Change Management. NONE means that the users not registered in the SQL ID table are never offered a choice to use Change Management when modifying DDL.

```
DB2 Admin -------------- CM - Manage ID Table --------------------- Row 1 of 3
Command ===>                                                  Scroll ===> CSR

Commands: SAVE
Line commands:  D - Delete  I - Insert

Change management default level setting : NONE


Sel SQL ID   Level    Oper
    *        *        *
--- -------- -------- ------
    PAOLOR1  OPTIONAL
    PAOLOR2  OPTIONAL
    PAOLOR3  OPTIONAL
****************************** END OF DB2 DATA ******************************
```

Figure 2-22 SQL ID table - Default set to NONE

If you want to restrict the access to Change Management to your DBAs, then you should setup the default level to NONE and register the SQL IDs of your DBAs. The level for the SQL IDs for the DBAs should be set to REQUIRED or OPTIONAL, depending on whether they should always register changes or not.

> **Note:** Make sure that you do not grant privileges on the SQL ID table in order to prevent unwanted updates.

2.3.3 Working with multiple Change Management Databases

We do not recommend using more than one set of Change Management tables within one subsystem. As you remember from *prefix*.SADBSAMP(ADBBIND), you are required to use the table creator of your Change Management tables in the BIND PACKAGE statements for Change Management. As a reminder, look at the example in Figure 2-23.

```
EDIT       ADB.V7R2M0.SADBSAMP(ADBBIND) - 01.02           Columns 00001 00072
Command ===>                                                 Scroll ===> CSR
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -   220 Line(s) not Displayed
020310  BIND PACKAGE(ADBL) -
020320       MEMBER(ADBCEST) -
020330       QUALIFIER(A72)      /*  THE QUALIFIER SHOULD MATCH THE OWNER */ -
020340       VALIDATE(BIND)      /*  OF THE CHANGE MANAGEMENT TABLES.     */ -
020350       ISOLATION(CS) -
020360       RELEASE(C) -
020370       ENCODING(EBCDIC)  -
020380       EXPLAIN(NO)
020390
020391  BIND PACKAGE(ADBL) -
020392       MEMBER(ADBCPRE) -
020393       QUALIFIER(A72)      /*  THE QUALIFIER SHOULD MATCH THE OWNER */ -
020394       VALIDATE(BIND)      /*  OF THE CHANGE MANAGEMENT TABLES.     */ -
020395       ISOLATION(CS) -
020396       RELEASE(C) -
020397       ENCODING(EBCDIC)  -
020398       EXPLAIN(NO)
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -   254 Line(s) not Displayed
****** *************************** Bottom of Data ****************************
```

Figure 2-23 Table creator of your CM database reflects the qualifier in your bind statements

In order to deal with multiple Change Management tables within the same subsystem, you have to bind a new package with a different collection. Additionally, the qualifier option from the new package has to be the creator from your second Change Management database. Because the consistency tokens of both packages are equal, you must use an additional DB2 Admin plan name pointing with its PKLIST entry to the new collection ID.

2.4 DB2 Security for Change Management

In this section we look at the security aspects when using Change Management. From a user's point of view, you might want to restrict access to Change Management because you want to make sure that just your DBAs have access to the DDL changes to be done in your system. Access to Change Management should be restricted to authorized users.

You can control access to Change Management either by your ADBL Clist or by DB2 security settings.

2.4.1 Access to Change Management based on your ADBL Clist

You can invoke DB2 Admin in such a way that Change Management is disabled. Refer to 2.1.6, "Invoking DB2 Admin" on page 20 to see how product invocation looks. If you do not want other parties to access DB2 Admin with Change Management, you must add the CMOWNER keyword with a dash. See Figure 2-24.

```
EDIT SYS1.ALLUSER.CLIST(ADB) - 01.06 Columns 00001 00072
Command ===> Scroll ===> CSR
****** **************************** Top of Data ******************************
000100 /** REXX ************************************************************/
000200 /* DB2 ADMIN AND COMPARE STARTER EXEC */
000300 /*********************************************************************/
000400 PARSE UPPER ARG PARM
000500
000600 IF POS(' PROD(',' 'PARM)=0 THEN
000700 PARM = PARM 'PROD(ADB.V7R2M0) PRODADD(GOC.V7R2M0)' ,
000800 'LIBPRE(SADB) LIBAPRE(SGOC) CMOWN(-)'
000900 ADDRESS 'ISPEXEC' 'SELECT MODE(FSCR) NEWAPPL(ADB)',
001000 'CMD(EX ''ADB.V7R2M0.SADBCLST(ADBL)'' '''PARM''')'
001100
001200 EXIT RC
****** **************************** Bottom of Data ***************************
```

Figure 2-24 Invoking DB2 Admin without access to Change Management

This invocation routine prevents Change Management from being executed. Change Management is not shown in the DB2 Admin Main Menu.

In comparison, Change Management is provided as shown in Figure 2-25. This invocation routine could be assigned to your DBAs only.

```
EDIT       SYS1.ONLYDBA.CLIST(ADB) - 01.14              Columns 00001 00072
Command ===>                                               Scroll ===> CSR
****** **************************** Top of Data ******************************
000100 /** REXX ************************************************************/
000200 /* DB2 ADMIN AND COMPARE STARTER EXEC                                */
000300 /*********************************************************************/
000400    PARSE UPPER ARG PARM
000500
000600    IF POS(' PROD(',' 'PARM)=0 THEN
000700      PARM = PARM 'PROD(ADB.V7R2M0) PRODADD(GOC.V7R2M0)' ,
000800                  'LIBPRE(SADB) LIBAPRE(SGOC)'
000900    ADDRESS 'ISPEXEC' 'SELECT MODE(FSCR) NEWAPPL(ADB)',
001000     'CMD(EX ''ADB.V7R2M0.SADBCLST(ADBL)'' '''PARM''')'
001100
001200    EXIT RC
****** **************************** Bottom of Data ***************************
```

Figure 2-25 Invoking DB2 Admin with access to Change Management

With these examples you can provide two separate ways of ADBL invocation routines. Please make sure that your product runtime libraries are protected from update as well, in such a way that no one can update *prefix*.SADBCLST(ADBL).

Note: As all of your GOC data sets are required for using Change Management as well, we do not recommend using RACF® data set authorization protection to restrict access to your environment in combination with values given for PRODADD and LIBAPRE. This is because of the dynamic allocation process of the data sets done by ADBL. If authorization for data sets is not given by RACF and the product tries to allocate those libraries, ADBL invocation fails.

2.4.2 Access to Change Management based on DB2 security

You can limit access based on DB2 security as well. For instance, suppose that you do not want to have more than one invocation routine as described in 2.4.1, “Access to Change Management based on your ADBL Clist” on page 39.

Assume that you want your DBAs to access Change Management, but, on the other hand, Change Management must not be used for application developers.

In any case you have to grant the DB2 Admin main plan ‘ADB’ to your DBAs and to your application staff as well.

Unless you do not grant DB2 security authorizations on your Change Management database, any CM function that requires access to the Change Management tables will fail. Users who own implicit access (like SYSADM or DBADM on the Change Management database) always have access to the Change Management functions.

In our example, PAOLO does not own any privileges on the Change Management database. In Figure 2-26, PAOLO tried to update the SQL ID table.

```
DB2 Admin -------------------- Change Management (CM) ------------------- 21:10
Option ===> 5


   1 - Manage changes                                    DB2 System: DB8A
   2 - Manage masks                                      DB2 SQL ID: PAOLO
   3 - Manage ignores
   4 - Manage versions
   5 - Manage ID table
   6 - Report changes
```

Figure 2-26 Choosing option 5 ‘Manage ID table’

See Figure 2-27 for an example of PAOLO’s authorization failure. Access is denied.

```
DB2 Admin --------------------- DB2 Error Display 1 --------------------- 21:08
Command ===>
Rollback done
   SQLCODE : -551                           DSNTIAR CODE :  0

DSNT408I SQLCODE = -551, ERROR:  PAOLO DOES NOT HAVE THE PRIVILEGE TO PERFORM
         OPERATION SELECT ON OBJECT A72.ADBCID
DSNT418I SQLSTATE   = 42501 SQLSTATE RETURN CODE
DSNT415I SQLERRP    = DSNXOSC SQL PROCEDURE DETECTING ERROR
DSNT416I SQLERRD    = -100  0  0  -1  0  0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD    = X'FFFFFF9C'  X'00000000'  X'00000000'  X'FFFFFFFF'
         X'00000000'  X'00000000' SQL DIAGNOSTIC INFORMATION
```

Figure 2-27 Access denied on the ID table

As the product uses dynamic SQL to access the tables, authorization is checked whenever you try to access these tables. In some situations you might have the feeling that access is being granted if a panel is shown, but this is just because that particular panel does not have access to DB2 tables. As soon as panels are involved that have to open a DB2 table of CM, security is checked and in case of not having appropriate rights, an SQL code is given, as you can see in Figure 2-27.

Note: Your application developers have access to DB2 Compare, which does not operate with DB2 packages and plans. Therefore, even if DB2 security prohibits access to DB2 Admin, DB2 Compare is shown in any case on your DB2 Admin main menu and it can be used. Refer also to “Access to your WSL libraries” in the following section.

2.4.3 Additional security aspects

Next we discuss some additional aspects in terms of security.

Access to your WSL libraries

In any case, every user can enter the **WSL** command in order to invoke the dialog to access your WSL libraries. If you want to prevent your users from modifying your WSLs, you must protect your WSL library within your security system.

DB2 privileges

A user is not allowed to use the CM process, because of the restrictions imposed by the methods described above, however the same user should not be granted DB2 privileges to alter the objects.

2.4.4 Update of changes

Be careful if you GRANT SELECT access to the Change Management tables. In this case, a userid has access to all available changes. Although it is just a DB2 SELECT privilege on a DB2 table, the user who owns the SELECT privilege could enter any of the available commands such as AN for *Analyze a change* or RN for *Run a change*. The user might indeed not own the privileges required to create the DDL given in a change, but there is always the possibility of overwriting JCL or a WSL, if those data sets are not protected. You must have them for your original changes.

Recommendation: Do NOT set up DB2 security for CM in a way that users can only read the registered changes. If you want to restrict users from access, you should not allow any access to your CM database.

2.4.5 Security based on collection IDs

In addition to the DB2 security for Change Management based on tables, you can disable access to the Change Management packages. The best way to do it is to look into your *prefix*.SADBSAMP first. You can find a comment for every Change Management package, as you can see in Figure 2-28.

```
VIEW       ADB.V7R2M0.SADBSAMP(ADBBIND) - 01.02          Columns 00001 00072
Command ===>                                               Scroll ===> CSR
****** ***************************** Top of Data ******************************
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -   245 Line(s) not Displayed
020418  BIND PACKAGE(ADBL) -
020419       MEMBER(ADBCRCR) -
020420       QUALIFIER(A72)      /*  THE QUALIFIER SHOULD MATCH THE OWNER */ -
020421       VALIDATE(BIND)      /*  OF THE CHANGE MANAGEMENT TABLES.     */ -
020422       ISOLATION(CS) -
020423       RELEASE(C) -
020424       ENCODING(EBCDIC)  -
020425       EXPLAIN(NO)
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -   236 Line(s) not Displayed
****** **************************** Bottom of Data ****************************
```

Figure 2-28 Identifying the CM packages

You should bind those packages that belong to Change Management to a new collection, for example, ADBLCM. You have to include this collection into your PKLIST entries in the DB2 Admin bind plan statements as well.

You then can explicitly GRANT the collection privilege PACKADM to ADBL for all users, but you must not GRANT the collection privilege PACKADM to ADBLCM.

2.4.6 Removing GRANTs

You might not be comfortable with having your changes generated by default with GRANT statements if your shop's security is based on an external alternative like RACF DSN@XACS or CA ACF2 for DB2. Currently, GRANT statements are always involved in your changes; you do not have access to a product function that could prohibit DB2 Admin and DB2 Compare from generating GRANTs.

However, if you do not want GRANT statements to appear during your change execution, IBM can provide special modifications relating to your *prefix*.SADBSLIB and *prefix*.SADBLLIB. Just ask your IBM representative how to get these special modifications.

Archived

3

First steps with DB2 Admin

In this chapter we describe the first steps you must take when you start using DB2 Admin. We also give you some hints and tips about how to use the primary functions of Version 7 Release 2 and include some of the lesser known functions.

Then we introduce ways to put the DB2 catalog information to good use for application developers, as well as database administrators in their everyday work.

We cover the following topics:

- Looking at database objects
- Creating new objects based on existing objects
- Working with a copy of the DB2 system catalog
- Showing data set information
- REVOKE and DROP impact
- Restarting the monitor
- Using work statement lists (WSLs)

For a detailed description of all functions and commands, refer to *DB2 Administration Tool for z/OS User's Guide and Reference Version 7 Release 2,* SC18-9552.

© Copyright IBM Corp. 2007. All rights reserved.

3.1 First time in DB2 Admin

Note: Once you have installed DB2 Admin and customized the installation as described in Chapter 2, “Customization” on page 15, you can activate the DB2 Admin function from its Main Menu shown in Figure 3-3.

3.1.1 Navigating the menus

You get to the DB2 Admin Menu panel from ISPF, either selecting the DB2 Admin entry in the Master Application Menu, or issuing a TSO command. Alternatively, you can get to the Menu after activating the DB2 Admin Tool Launchpad.

When you start DB2 Admin from the Master Application Menu ISPF panel, you usually see the DB2 Admin panel that shows the active DB2 subsystems on the z/OS system where you logged onto TSO. On our system, it looks like Figure 3-1.

```
DB2 Admin ------------------- Active DB2 Systems ------------------ Row 1 of 2
Command ===>                                                Scroll ===> PAGE


This is a list of the active DB2 systems on this MVS system.

Enter:
DB2 system name ===>              Retain DB2 system name ===> Y  (Y/N)

Or select the one you wish to use, or press END to exit.

Sel DB2 System Description                                              Group
--- ---------- ------------------------------------------------------- -----
S   DB8A       ITSO DB2 V8 Subsystem
    DB9B       ITSO DB2 V9 Subsystem
****************************** Bottom of data ******************************
```

Figure 3-1 Active DB2 Systems panel

There are two active DB2 subsystems on our z/OS. We work with the local DB2 system “DB8A”, which has DB2 Admin installed. Type **S** next to DB8A to select it, or type in `DB8A` in the DB2 system name field, and press Enter.

If there is only a single DB2 subsystem on the z/OS image, you normally do not see this panel. Also, you might not see this panel if it has been customized in your shop to allow you to work with only one local DB2 subsystem. The customization process is described in *DB2 Administration Tool for z/OS User’s Guide and Reference Version 7 Release 2,* SC18-9552.

In these cases, you go immediately to the DB2 Admin Menu panel shown in Figure 3-3.

The DB2 Admin Tool Launchpad (see Figure 3-2) is a convenient way to launch any one of the installed IBM DB2 Tools that have an ISPF interface directly from a centralized panel (thus avoiding having to go back to a list in the ISPF Master Applications Menu).

The tailoring of this panel is described in Chapter 3. “Using the DB2 Admin Launchpad” in *DB2 Administration Tool for z/OS User’s Guide Version and Reference Version 7 Release 2*, SC18-9552. The other IBM DB2 Tools have similar documentation for adding their entries to the Launchpad.

```
------------------------- DB2 Tools Launchpad ----------------- Row 1 from 11
  Command ===>                                                  Scroll ===> CSR


  Specify DB2 SSID (opt) ===>  DB8A  (Enter '?' for a list of active SSIDs)

  Select the DB2 tool you wish to launch or enter its code in the command line.

  Sel Code  Tool Name                                        Rel  Prog No.
      ---   ------------ ADMINISTRATION TOOLS -------------  ---  --------
      ADM   DB2 Administration Tool                          720  5697-L90
      OC    DB2 Object Comparison Tool                       720  5697-L40
      ---   --------- PERFORMANCE MANAGEMENT TOOLS --------  ---  --------
      PE    OMEGAMON XE for DB2 Performance Expert on z/OS   410  5655-Q07
****************************** Bottom of data ********************************
```

Figure 3-2 DB2 Admin Launchpad

As listed in the Launchpad, in our case we have DB2 Admin, DB2 Compare, and OMEGAMON® XE for DB2 PE. Once you specify the SSID of the DB2 system you want to connect to, you get to the DB2 Admin Menu as shown in Figure 3-3.

```
 DB2 Admin -------------- DB2 Administration Menu 7.2.0 ------------------ 14:49
 Option ===>

    1 - DB2 system catalog                          DB2 System: DB8A
    2 - Execute SQL statements                      DB2 SQL ID: PAOLOR3
    3 - DB2 performance queries                     Userid    : PAOLOR3
    4 - Change current SQL ID                       DB2 Rel   : 815
    5 - Utility generation using LISTDEFs and TEMPLATEs
    P - Change DB2 Admin parameters
   DD - Distributed DB2 systems
    E - Explain
    Z - DB2 system administration
   SM - Space management functions
    W - Manage work statement lists
    X - Exit DB2 Admin
   CC - DB2 catalog copy version maintenance
   CM - Change management

 Interface to other DB2 products and offerings:
    C  DB2 Object Comparison Tool
    I  DB2I
```

Figure 3-3 DB2 Admin Menu

The top right of the panel shows the DB2 subsystem you are working on and the TSO user connected. The top left of the panel shows the native DB2 Admin functions you can choose. The bottom of the panel shows additional DB2 products that can be invoked from this menu. In this sample you see two other functions. The first one is the DB2 Object Comparison Tool, also known as DB2 Object Compare, or just *DB2 Compare*, which we used for our Change Management activities. The second one is *DB2 I*, the classic interactive DB2 SPUFI interface.

The functions have been added by the DB2 Admin customization process, described in Chapter 2, “Customization” on page 15. By following this process, you can also add other IBM Data Management Tools to this menu, if necessary.

3.1.2 Customizing your DB2 Admin parameters

Before you start using DB2 Admin, you should customize the DB2 Admin parameters. To do so, enter **P** on the DB2 Admin Menu panel. This brings you to the Change DB2 Admin Parameters panel shown in Figure 3-4.

```
DB2 Admin ---------------- DB2 Change DB2 Admin Settings ---------------- 15:01
Option ===>

   0 - Change ISPF settings                              DB2 System: DB8A
   1 - Change colors and highlights                      DB2 SQL ID: PAOLOR3
   2 - Change DB2 Admin defaults
   P - Change/allocate print data set
  BP - Change batch parameters
  PR - Change prompt options
   M - Change migrate settings
```

Figure 3-4 Change DB2 Admin Parameters panel

Using options **0** and **1**, you can now change the ISPF parameters and the DB2 Admin panels. Normally, you do not have to change these parameters and can use the default settings. We did not change the defaults.

Use option **2** to change various parameters that affect the execution of DB2 Admin.

Note: If you decide to change any of the DB2 Admin Defaults by using option 2, you have to change the field *Reset to Def. at Startup* to *NO* on this panel if you want these changes to be persistent across subsequent sessions.

Changing these settings is optional, however you should customize the next three settings.

In option **M** the migrate settings can be changed. Space information is gathered if necessary prior to building the migrate job(s). If YES is specified (default), space information is displayed on the panel, *Migrate Table Spaces*.

Allocating the print data set

To allocate the print data set, select option **P**, which takes you to the panel shown in Figure 3-5.

```
DB2 Admin -------------- Change/Allocate Print Data Set ----------------- 14:01
Option ===>

Enter data set name and disposition:
  Data set name ===> 'PAOLOR3.PRINT'
  Disposition   ===> NEW        (NEW,OLD,MOD,FREE)

For a NEW data set enter:
  Lrecl         ===> 132        (8-32760)
  Block size    ===> 6204       (0-32760)
  Format        ===> F          (Fixed or Variable)
  Space units   ===> TRKS       (Tracks, Cylinders or Blocks)
  Primary space ===> 10         (Default 1)
  Sec. space    ===> 1          (Default 1)
  Unit type     ===>            (Default SYSDA)
```

Figure 3-5 Allocating the print data set

On this panel you can select the various data set options to allocate your print data set.

Attention: Once you have created a print data set, you should change the disposition from NEW, otherwise you are not able to print output to the data set. As can be seen below, we changed ours to OLD.

```
DB2 Admin -------------- Change/Allocate Print Data Set ----------------- 14:21
Option ===>

Enter data set name and disposition:
  Data set name ===> 'PAOLOR3.PRINT'
  Disposition   ===> OLD        (NEW,OLD,MOD,FREE)

For a NEW data set enter:
  Lrecl         ===> 132        (8-32760)
  Block size    ===> 6204       (0-32760)
  Format        ===> F          (Fixed or Variable)
  Space units   ===> TRKS       (Tracks, Cylinders or Blocks)
  Primary space ===> 10         (Default 1)
  Sec. space    ===> 1          (Default 1)
  Unit type     ===>            (Default SYSDA)
```

Figure 3-6 Changing the disposition of the print data set

Printing to the data set is described fully in “Printing output to the print data set” on page 69.

Customizing your batch parameters

To change the batch parameters, choose option **BP** on the panel shown in Figure 3-4 on page 48, which takes you to the panel shown in Figure 3-7.

Archived

```
DB2 Admin -------------- DB8A Batch Job Utility Parameters -------------- 15:21
Command ===>


Generate Job Card . ===> YES  (Yes/No)                    DB2 System: DB8A
 Job cards:                                               DB2 SQL ID: PAOLOR3
  ===> //PAOLOR3R JOB (999,POK),REGION=0M,MSGCLASS=X,
  ===> //         MSGLEVEL=(1,1),NOTIFY=&SYSUID,
  ===>
  ===>
  ===>
 Generate Job CLASS ===> YES  (Yes/No)     JOB CLASS . . . . . ===> A

JOBPARM:
  ===> /*JOBPARM S=SC63
  ===>
  ===>
  ===>

ADBTEP2: Restart . .===>  YES (Yes/No)
         Maxerrors .===>   -1 (-1 to 99)

Space parameters:
  Unit name         ===> SYSDA
  Space unit  . . . ===> TRK      (BLK, TRK, CYL or 4096-32760)
  Max Primary . . . ===> 65535    (In above units or 99999999 or blank)
                                   In KB: 3145680
  Max DASD  . . . . ===> 65535    (In above units. Allocations beyond this
                                   are sent to tape) In KB: 3145680
  Tape Unit . . . . ===> TAPE      (Unit for tape if size is greater
                                   than Max DASD)
Default space allocation if unable to calculate:
  Primary alloc . . ===> 30       (in above units)
  Secondary alloc . ===> 30       (in above units)
```

Figure 3-7 Batch parameters panel

On this panel you can change the job card and the JES2 JOBPARM control statement.

You can also select whether or not to generate the job card and/or job CLASS to override the default installation values.

Attention: The last line of the job card should end with a comma if you set *JOB CLASS* to *Y*, because DB2 Admin adds an additional line to the job card for the job CLASS.

If JOBPARM is not specified on this panel; DB2 Admin adds a line in the generated job card that is derived from the installation specified JOBPARM.

DB2 Admin has a function that monitors the execution of jobs running with the batch execution program *ADBTEP2*. This function allows the restart of DB2 Admin jobs if they have terminated abnormally; see also 3.8, “Restarting the monitor” on page 88. If you do not want this capability installed, type **N** in the field “ADBTEP2 restart”. In this case, a **RESTART(NO)** parameter is generated for each job. We recommend using the default, which is **YES**.

The value for *maxerrors* specifies the number of DSN commands that can fail before the ADBTEP2 batch job is stopped:

- **-1** All DSN errors are ignored. The job is not stopped for any error.
- **0** No DSN errors all ignored. The job is stopped on the first error.
- **1-99** The specified number of DSN errors are ignored.
  The job is stopped on the nth + 1 error.

The *Default unit name* is used for the allocation of new data sets.

The *Default space allocation* is used for allocating image copy and unload data sets if RUNSTATS or STOSPACE has not been run, and the tool has no information about the current size of the objects. Otherwise, these values are derived from the DB2 system catalog.

Attention: If the **BP** command is issued when connected to a remote subsystem, then all specifications should be defined for use when connected to a remote subsystem. If the **BP** command has never been issued when connected to a remote subsystem, then the specification for the local system is used.

Setting the Prompt Options

Entering **PR** on the panel shown in Figure 3-4 on page 48, brings you to the Prompt Options panel, shown in Figure 3-8.

```
DB2 Admin ------------------------ Prompt Options ----------------------- 14:51
 Option ===>

 Change one or more options below. Prompt before executing:

   Definition SQL (CREATE, DROP, ALTER, RENAME,.)  ===> YES (Yes or No)

   Authorization SQL (GRANT and REVOKE)            ===> YES (Yes or No)

   Update SQL (INSERT, UPDATE, DELETE)             ===> YES (Yes or No)

   DSN commands (BIND, REBIND and FREE)            ===> YES (Yes or No)

   DB2 commands (START, STOP, ALTER, SET)          ===> YES (Yes or No)
```

Figure 3-8 Prompt Options panel

On this panel, you can set the DB2 Admin prompt options. The execution of an SQL statement, a DSN command, or a DB2 command is deferred if the prompt for these commands or statements is set to **YES**. Specifying **YES** activates the prompting for the specific function. This means that before the function is executed, you are asked what you want to do. The available options are:

- Execute the statement immediately.
- Edit the statement.
- Create a batch job to execute the statement.
- Add the statement to a work statement list (WSL), if the function supports WSL.
  More details about work statement lists can be found in 3.9, “Using work statement lists” on page 91.

Note: The default option is *no prompting*. If you want to be prompted before executing statements, specify **YES**.

A sample prompt panel for statement execution is shown in Figure 3-9 with the four possible options.

```
DB2 Admin --------------- DB8A Statement Execution Prompt --------------- 17:18
Option ===>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
   1 - Execute the statement
   2 - Edit the statement
   3 - Create a batch job with the statement
   4 - Add the statement to the work statement list
 CAN - Cancel
 Work statement list dsn  ===> 'PAOLOR3.WORKLIST'
 Work statement list name ===> GRANTADB      Action ===> A (Append or Replace)

Statement that is about to be executed (first 28 lines):
GRANT EXECUTE
  ON PLAN ADBTEP2
  TO  PUBLIC
```

Figure 3-9 Statement Execution Prompt

Here are the four possible options:

- Option **1** executes the statement immediately.
- Option **2** shows you the statement or command in EDIT mode and you can edit it. Pressing PF3 shows you the panel in Figure 3-9 again.
- Option **3** generates a batch job to execute the statement using the batch execution program ADBTEP2. The SYSIN that ADBTEP2 uses is sometimes referred to as the *Batch Statement List* (to differentiate if from a *Work Statement List* (WSL). WSLs are treated in more detail in Chapter 3.9, “Using work statement lists” on page 91.
- Option **4** adds the statement to a work statement list. You have to enter a data set name for the work statement list and a member name for the list itself. If the data set does not exist, DB2 Admin creates it for you.

  Use work statement action **A** to append the statement to an existing work statement list member or action **R** to replace the member prior to writing the statement.

Note: You can use the primary command **PROMPT ON** or **PROMPT OFF** to enable or disable prompting from each DB2 Admin panel. You can also get to the Prompt Option panel at any time by entering the **PROMPT** command. In addition, you can make **PROMPT YES** be the default by use of the tag in the ADB2CUST EXEC mentioned at 2.1.5, “Running the customization EXEC” on page 19.

3.2 Looking at database objects

By selecting option **1** on the DB2 Administration Menu, shown in Figure 3-3 on page 47, you can initiate a search for DB2 objects or DB2 authorizations in the DB2 system catalog. The System Catalog object options panel shown in Figure 3-10 is presented.

```
DB2 Admin --------------------- DB8A System Catalog --------------------- 17:43
 Option ===> AO


 Object options:                                          DB2 System: DB8A
   AO - Authorization options                             DB2 SQL ID: PAOLOR3
    G - Storage groups                 P - Plans
    D - Databases                      L - Collections
    S - Table spaces                   K - Packages
    T - Tables, views, and aliases     M - DBRMs
    V - Views                          H - Schemas
    A - Aliases                        E - User defined data types
    Y - Synonyms                       F - Functions
    X - Indexes                        O - Stored procedures
    C - Columns                        J - Triggers
    N - Constraints                    Q - Sequences
   DS - Database structures          DSP - DS with plans and packages

 Enter standard selection criteria (Using a LIKE operator, criteria not saved):
 Name     ===>                    >  Grantor  ===>          >
 Owner    ===>          >            Grantee  ===>          >
 In D/L/H ===>                    >
 And/or other selection criteria (option xC shows you columns for option x)
 Column   ===>                    >  Operator ===>        Value ===>
```

Figure 3-10 System Catalog panel - object options

On this panel you see the option codes you can enter on the command line to work on DB2 objects. Choosing option **AO** takes you to System Catalog authorizations panel (Figure 3-11), which shows the option codes for operations with various types of DB2 authorizations.

```
DB2 Admin --------------------- DB8A System Catalog --------------------- 17:45
Option ===>


Authorization options:                                    DB2 System: DB8A
  OO - Object options                                     DB2 SQL ID: PAOLOR3
  GA - Storage group auths           PA - Plan authorizations
  DA - Database authorizations       LA - Collection authorizations
  SA - Table space authorizations    KA - Package authorizations
  TA - Table authorizations          HA - Schema authorizations
  VA - View authorizations           EA - User defined data type authorizations
  CA - Column authorizations         FA - Function authorizations
  ZA - System authorizations         OA - Stored procedure authorizations
  UA - User authorizations           QA - Sequence authorizations
  RA - Resource authorizations


Enter standard selection criteria (Using a LIKE operator, criteria not saved):
Name     ===>                    >  Grantor  ===>          >
Owner    ===>          >            Grantee  ===>          >
In D/L/H ===>                    >
And/or other selection criteria (option xC shows you columns for option x)
Column   ===>                    >  Operator ===>        Value ===>
```

Figure 3-11 System Catalog panel - authorization options

When you are looking at DB2 objects, you might want to follow the relationships between objects. Figure 3-12 shows the main links between objects supported by DB2 Admin. This means that if you find an object, you can look for various dependent objects or you can look for the parent object of the selected object as indicated by the arrows in the figure.

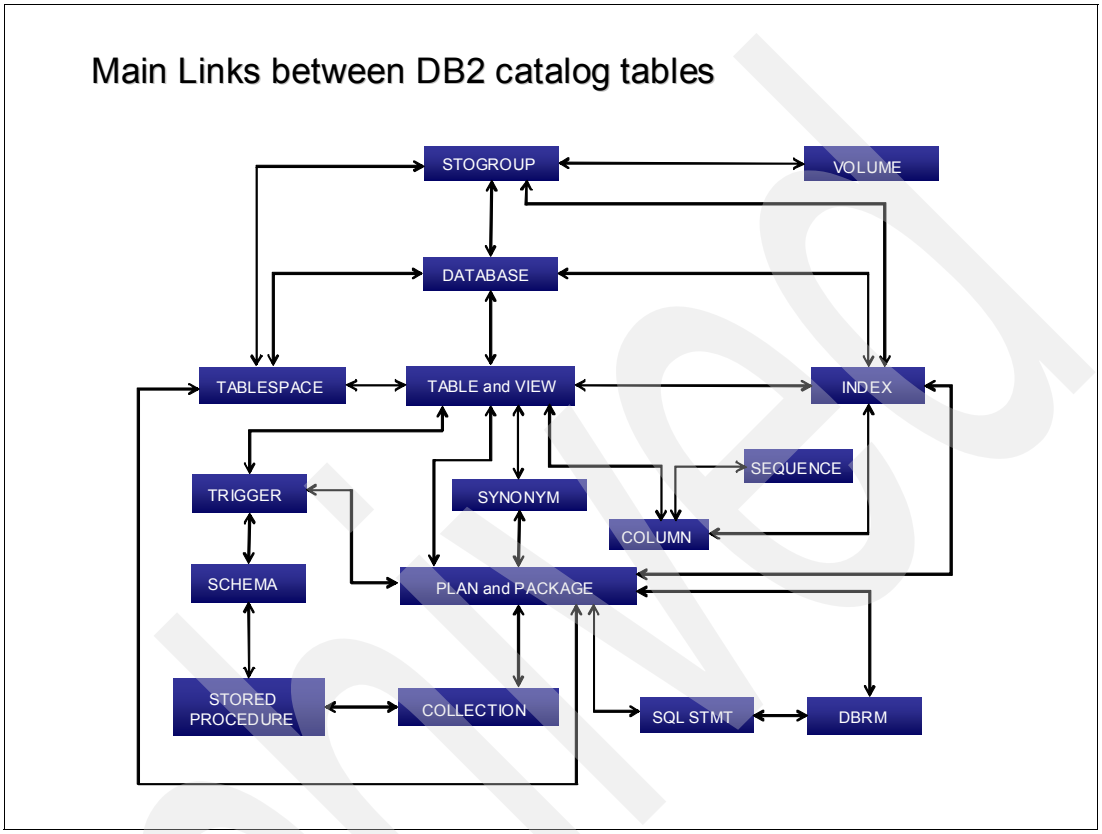

Figure 3-12 Main links between DB2 catalog tables

3.2.1 Navigating the DB2 system catalog

Next, we show how to use DB2 Admin if you are searching for objects in the DB2 catalog.

Assume that we are looking for table spaces in our **DSG** databases owned by the user ID **PAOLOR3**.

> **Note:** You can save or not save your search criteria between DB2 Admin sessions by using the **SAVE ON** and **SAVE OFF** primary commands. When the **SAVE ON** command is active, the text `criteria saved` appears on the System Catalog panel. With **SAVE ON**, the search criteria is restored when you re-enter a DB2 Admin session.

In the DB2 System Catalog object options panel, you choose option **S** to display the table spaces and enter the name of the database and the database owner ID as selection criteria, as shown in Figure 3-13.

```
DB2 Admin --------------------- DB8A System Catalog --------------------- 19:20
Option ===> S


Object options:                                          DB2 System: DB8A
  AO - Authorization options                             DB2 SQL ID: PAOLOR3
   G - Storage groups                 P - Plans
   D - Databases                      L - Collections
   S - Table spaces                   K - Packages
   T - Tables, views, and aliases     M - DBRMs
   V - Views                          H - Schemas
   A - Aliases                        E - User defined data types
   Y - Synonyms                       F - Functions
   X - Indexes                        O - Stored procedures
   C - Columns                        J - Triggers
   N - Constraints                    Q - Sequences
  DS - Database structures          DSP - DS with plans and packages

Enter standard selection criteria (Using a LIKE operator, criteria saved):
Name     ===>                    >  Grantor  ===>          >
Owner    ===> PAOLOR3% >            Grantee  ===>          >
In D/L/H ===> DSG%               >
And/or other selection criteria (option xC shows you columns for option x)
Column   ===>                    >  Operator ===>        Value ===>
```

Figure 3-13 Searching table spaces in database DSG% and database owner PAOLOR3%

You see a comment before the search criteria that an SQL LIKE operator is to be used. DB2 Admin generates an SQL statement that searches the DB2 catalog for objects that match your search criteria. For each field, DB2 Admin generates an SQL LIKE operator and adds a percent sign (**%**) at the end of your search string.

You can turn off the LIKE search by entering the **LIKE OFF** primary command. In this case, the EQUAL search is used and the search criteria must reflect the exact search value. **LIKE ON** switches back to using the LIKE search again.

Even though you have LIKE OFF, you can still use the **%** (or *****) sign to do a LIKE search. So in the example in Figure 3-13 with LIKE OFF, you would have to enter PAOLOR3* (or PAOLOR3%) to do a LIKE search.

Archived

The results of the DB2 catalog search are shown in Figure 3-14.

```
DB2 Admin ------------------ DB8A Table Spaces ------------- Row 1 to 16 of 16
Command ===>                                                  Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line commands

Select Name     DB Name   Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
       *        *             * *      * * * * *      *           *     * * *
------ -------- -------- ------ ------ - - - - - ------ ----------- ------ - -
       GLWS001  DSG24D02      0 BP15   A N A N Y      7          -1      4   Y
       GLWS002  DSG24D02      0 BP15   A N A N Y      4          -1      4   Y
       GLWSDPT  DSG24D02      0 BP15   A N A N Y      1          -1     32   Y
       GLWSEMP  DSG24D02      4 BP15   A N A N Y      1          -1      0   Y
       GLWSEPA  DSG24D02      0 BP15   A N A N Y      1          -1     64   Y
       GLWSPJA  DSG24D02      0 BP15   A N A N Y      1          -1     32   Y
       GLWSPRJ  DSG24D02      0 BP15   A N A N Y      1          -1     32   Y
       GLWSSPL  DSG24D02      4 BP15   A N A N Y      1          -1      0   Y
       GLWS001  DSG24D03      0 BP15   A N A N Y      7          60      4   Y
       GLWS002  DSG24D03      0 BP15   A N A N Y      4         360      4   Y
       GLWSDPT  DSG24D03      0 BP15   A N A N Y      1         360     32   Y
       GLWSEMP  DSG24D03      4 BP15   A N A N Y      1         144      0   Y
       GLWSEPA  DSG24D03      0 BP15   A N A N Y      1         360     64   Y
       GLWSPJA  DSG24D03      0 BP15   A N A N Y      1         360     32   Y
       GLWSPRJ  DSG24D03      0 BP15   A N A N Y      1         360     32   Y
       GLWSSPL  DSG24D03      4 BP15   A N A N Y      1         432      0   Y
****************************** END OF DB2 DATA ******************************
```

Figure 3-14 Result of searching table spaces in databases DSG%

In the result, you can drill up or down in the DB2 system catalog and look for *child* or *parent* objects. You also have the possibility to get more information on a specific object by entering a line command. The valid line commands for each object are shown in the upper part of the result panel. Usually not all commands can be shown there because of the limited amount of space available. Typing a question mark (**?**) as a line command in front of an object shows all valid line commands for that object.

> **Note:** Selecting option **T** on the System Catalog panel shows results for all tables, views, and aliases. If you only want to list tables, then option **TT** can be used.

Using search arguments

The line below the column name is the search arguments line. For example, in Figure 3-14, this the line containing an asterisk (*****) below Select Name, DB Name..., For every column, you can overtype the ***** with a search argument. Entering a search argument limits the result of the presented list.

If you only want to see the table spaces that belong to the database *DSG24D03*, enter **DSG24D03** in the search arguments line under the column, DB Name, in the panel shown in Figure 3-14. The result is shown in Figure 3-15.

Note: Make sure the value you enter is properly aligned. In our case, *DB Name* is a character field. Therefore it must be aligned to the left. DB2 Admin helps you by placing an asterisk (*) indicating where you should align to.

```
DB2 Admin ------------------ DB8A Table Spaces ----------------- Row 9 from 16
Command ===>                                                   Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line commands

Select Name     DB Name   Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
       *        DSG24D03      * *      * * * * *      *           *      * * *
------ -------- -------- ------ ------ - - - - - ------ ----------- ------ - -
       GLWS001  DSG24D03      0 BP15   A N A N Y      7          60      4   Y
       GLWS002  DSG24D03      0 BP15   A N A N Y      4         360      4   Y
       GLWSDPT  DSG24D03      0 BP15   A N A N Y      1         360     32   Y
       GLWSEMP  DSG24D03      4 BP15   A N A N Y      1         144      0   Y
       GLWSEPA  DSG24D03      0 BP15   A N A N Y      1         360     64   Y
       GLWSPJA  DSG24D03      0 BP15   A N A N Y      1         360     32   Y
       GLWSPRJ  DSG24D03      0 BP15   A N A N Y      1         360     32   Y
       GLWSSPL  DSG24D03      4 BP15   A N A N Y      1         432      0   Y
****************************** END OF DB2 DATA ******************************
```

Figure 3-15 Result of showing table spaces which belong to database DSG24D03

Note: DB2 Admin normally uses wild carding to find DB2 objects, by using the LIKE predicate. You can use the primary command **SEARCH** or **SARG** to find an exact value instead.

Let us assume that this time we only want to see table spaces with four tables. If you are familiar with the DB2 system catalog tables, you can use the **SEARCH** primary command on the panel shown in Figure 3-14. After pressing the Enter key, the panel shown in Figure 3-16 is displayed.

```
DB2 Admin ------------------ DB8A Search fields ------------ Row 1 to 36 of 41
Command ===>                                                  Scroll ===> CSR

                          DB2    Srch
Select Column Name        Col No Oper Search Value
       *                     * *    *
------ ------------------ ------ ---- ----------------------------------------
       NAME                    1      *
       CREATOR                 2      *
       DBNAME                  3      *
       DBID                    4      *
       OBID                    5      *
       PSID                    6      *
       BPOOL                   7      *
       PARTITIONS              8      *
       LOCKRULE                9      *
       PGSIZE                 10      *
       ERASERULE              11      *
       STATUS                 12      *
       IMPLICIT               13      *
       NTABLES                14 =    4
       NACTIVE                15      *
       DSETPASS               16      *
```

Figure 3-16 Display search fields for a DB2 system catalog table

In this panel, the DB2 system catalog columns that describe the selected object are displayed. In this case they are the columns of the SYSIBM.SYSTABLESPACE table. Now you can enter the display and search arguments.

We enter an = as search operator and **4** for the search value for the field NTABLES, which contains the number of tables within a table space. Make sure the value you enter is properly aligned. In this case NTABLES is an integer field. Therefore it must be aligned to the right. DB2 Admin helps you by placing a ***** where you should align to.

There is only one table space in our database that contains exactly four tables, as the result in Figure 3-17 shows.

```
DB2 Admin ----------------- DB8A Table Spaces ------------------ Row 2 from 8
Command ===>                                                  Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line commands

Select Name     DB Name   Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
       *        *             * *      * * * * *      4           *      * * *
------ -------- -------- ------ ------ - - - - - ------ ----------- ------ - -
       GLWS002  DSG24D03      0 BP15   A N A N Y      4         360      4   Y
******************************** END OF DB2 DATA ********************************
```

Figure 3-17 Table spaces with four tables

Note: By using the **SEARCH** command, you turn off wild carding for only one query, whereas by using the **LIKE OFF** command, the wild carding is suspended for all queries until the LIKE processing is reset by the **LIKE ON** command.

Re-sort a result list

If you want to change the order of a list, you can use the **SORT** primary command that can be entered on any list panel. After pressing the Enter key, the next panel shows the columns of the underlying DB2 catalog table. You can enter a “Sort column No” to change the “DB2 column No” and choose whether this column should be ordered ascending or descending.

If, as in the sample list shown in Figure 3-14 on page 56, we want to change the order of the list to be by the NACTIVE column in descending order, you specify the options as shown in Figure 3-18.

```
DB2 Admin ------------------- DB8A Sort fields ------------- Row 1 to 15 of 15
Command ===>                                                 Scroll ===> CSR

Commands: SAVE  DELETE
Line commands:  n - Sort col no  A - Ascending  D - Descending

                                                Table  Sort   Asc/
Select Column Name          Column Header Name  Col No Col no Desc
       *                    *                     *      * *
------ -------------------- ------------------- ------ ------ ----
       NAME                                        1      0 ASC
       CREATOR                                     2      0 ASC
       DBNAME                                      3      0 ASC
       DBID                                        4      0 ASC
       OBID                                        5      0 ASC
       PSID                                        6      0 ASC
       BPOOL                                       7      0 ASC
       PARTITIONS           PARTS                  8      0 ASC
       LOCKRULE             L                      9      0 ASC
       PGSIZE                                     10      0 ASC
       ERASERULE            E                     11      0 ASC
       STATUS               S                     12      0 ASC
       IMPLICIT             I                     13      0 ASC
       NTABLES              TABLES                14      0 ASC
       NACTIVE              ACTPAGES              15      1 DESC
****************************** END OF DB2 DATA ******************************
```

Figure 3-18 Changing the sort order

The results of this option can be seen in Figure 3-19.

```
DB2 Admin ------------------ DB8A Table Spaces --------------- Row 1 to 8 of 8
Command ===>                                                  Scroll ===> CSR
Sort performed
Commands: GRANT  MIG  DIS  STA  STO
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line commands

Select Name     DB Name   Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
       *        *             * *      * * * * *      *          *     * * *
------ -------- -------- ------ ------ - - - - - ------ ----------- ------ - -
       GLWSSPL  DSG24D03      4 BP15   A N A N Y      1         432      0   Y
       GLWSDPT  DSG24D03      0 BP15   A N A N Y      1         360     32   Y
       GLWSEPA  DSG24D03      0 BP15   A N A N Y      1         360     64   Y
       GLWSPJA  DSG24D03      0 BP15   A N A N Y      1         360     32   Y
       GLWSPRJ  DSG24D03      0 BP15   A N A N Y      1         360     32   Y
       GLWSO02  DSG24D03      0 BP15   A N A N Y      4         360      4   Y
       GLWSEMP  DSG24D03      4 BP15   A N A N Y      1         144      0   Y
       GLWSO01  DSG24D03      0 BP15   A N A N Y      7          60      4   Y
****************************** END OF DB2 DATA ******************************
```

Figure 3-19 Table spaces ordered by active pages descending

Save SORT criteria

You can save a sort sequence for a panel using the **SAVE** command. The saved sort sequence for each panel is saved in an ISPF table named ADBSORT in your ISPF profile data set. The saved sort sequence is used next time DB2 Admin displays the same panel.

Use the **DELETE** command to delete a saved sort sequence.

Note: You also can use the **SORT** command in conjunction with the column names or a list of column names separated by a space. For example, you can type **SORT NTABLES NACTIVE** to order the list by number of tables and number of active pages. If you want to sort on a single column, you can enter the **SORT** command and place the cursor in the column you want to sort on and press Enter.

Archived

You can also use SORT to change the order of the results by specifying the initial of one of the columns appearing in the panel containing the results of your search. For instance, in the panel reported in Figure 3-20, we specify **sort n** to sort the result by table space **n**ame.

```
DB2 Admin ------------------ DB8A Table Spaces ------------- Row 1 to 16 of 16
Command ===> sort n                                            Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line commands

Select Name     DB Name   Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
       *        *             * *      * * * * *      *           *      * * *
------ -------- -------- ------ ------ - - - - - ------ ----------- ------ - -
       GLWS001  DSG24D02      0 BP15   A N A N Y      7          -1      4   Y
       GLWS002  DSG24D02      0 BP15   A N A N Y      4          -1      4   Y
       GLWSDPT  DSG24D02      0 BP15   A N A N Y      1          -1     32   Y
       GLWSEMP  DSG24D02      4 BP15   A N A N Y      1          -1      0   Y
       GLWSEPA  DSG24D02      0 BP15   A N A N Y      1          -1     64   Y
       GLWSPJA  DSG24D02      0 BP15   A N A N Y      1          -1     32   Y
       GLWSPRJ  DSG24D02      0 BP15   A N A N Y      1          -1     32   Y
       GLWSSPL  DSG24D02      4 BP15   A N A N Y      1          -1      0   Y
       GLWS001  DSG24D03      0 BP15   A N A N Y      7          60      4   Y
       GLWS002  DSG24D03      0 BP15   A N A N Y      4         360      4   Y
       GLWSDPT  DSG24D03      0 BP15   A N A N Y      1         360     32   Y
       GLWSEMP  DSG24D03      4 BP15   A N A N Y      1         144      0   Y
       GLWSEPA  DSG24D03      0 BP15   A N A N Y      1         360     64   Y
       GLWSPJA  DSG24D03      0 BP15   A N A N Y      1         360     32   Y
       GLWSPRJ  DSG24D03      0 BP15   A N A N Y      1         360     32   Y
       GLWSSPL  DSG24D03      4 BP15   A N A N Y      1         432      0   Y
****************************** END OF DB2 DATA ******************************
```

Figure 3-20 Specifying table spaces to be sorted by name

Archived

The result of this sort is shown in Figure 3-21. The sort can be specified by indicating the first letter of any column in the result set in the panel: Name (n), DB Name (d), Parts (p), Bpool (b) etc. The letters can also be combined as in **sort n d** to indicate sorting by table space name and database name.

```
DB2 Admin ------------------ DB8A Table Spaces ------------- Row 1 to 16 of 16
Command ===>                                                  Scroll ===> CSR
Sort performed
Commands: GRANT  MIG  DIS  STA  STO
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line commands

Select Name     DB Name   Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
       *        *             * *      * * * * *      *           *      * * *
------ -------- -------- ------ ------ - - - - - ------ ----------- ------ - -
       GLWSDPT  DSG24D02      0 BP15   A N A N Y      1          -1     32   Y
       GLWSDPT  DSG24D03      0 BP15   A N A N Y      1         360     32   Y
       GLWSEMP  DSG24D02      4 BP15   A N A N Y      1          -1      0   Y
       GLWSEMP  DSG24D03      4 BP15   A N A N Y      1         144      0   Y
       GLWSEPA  DSG24D02      0 BP15   A N A N Y      1          -1     64   Y
       GLWSEPA  DSG24D03      0 BP15   A N A N Y      1         360     64   Y
       GLWSPJA  DSG24D02      0 BP15   A N A N Y      1          -1     32   Y
       GLWSPJA  DSG24D03      0 BP15   A N A N Y      1         360     32   Y
       GLWSPRJ  DSG24D02      0 BP15   A N A N Y      1          -1     32   Y
       GLWSPRJ  DSG24D03      0 BP15   A N A N Y      1         360     32   Y
       GLWSSPL  DSG24D02      4 BP15   A N A N Y      1          -1      0   Y
       GLWSSPL  DSG24D03      4 BP15   A N A N Y      1         432      0   Y
       GLWS001  DSG24D02      0 BP15   A N A N Y      7          -1      4   Y
       GLWS001  DSG24D03      0 BP15   A N A N Y      7          60      4   Y
       GLWS002  DSG24D02      0 BP15   A N A N Y      4          -1      4   Y
       GLWS002  DSG24D03      0 BP15   A N A N Y      4         360      4   Y
****************************** END OF DB2 DATA ******************************
```

Figure 3-21 Table spaces sorted by name - Result set

Removing lines from a result list

Primary commands, such as **UTIL**, **GRANT** or **GEN,** generate commands or statements for all objects in a result list. Sometimes you don't want commands or statements for the complete list. You can then use a minus sign (**-**) as a line command to remove that line from the result list. Generation for that line does not take place. If you want to undo this change, you can use the ISPF primary command **REFRESH** and you get the complete list as it was before issuing the **-** command.

Using the BROWSE command

Catalog display panels are built from ISPF tables, where all results from catalog searching are stored. Unfortunately today, the ISPF Table Display Service does not allow left and right scrolling of a row, and the display of a row is limited to 80 columns. If your result row is longer than 80 columns, you can use the DB2 Admin **BROWSE** primary command to see the rest of the row.

Figure 3-22 shows the use of the **BR(OWSE)** command to see all columns of the objects listed.

> **Note:** If BROWSE (or BR) is defined as an ISPF command, then using the command inside Admin invokes the ISPF browse panel. To prevent this and use the Admin BROWSE command, just prefix it with the “greater than” sign (>), like **>BR**.

```
DB2 Admin --------------- DB8A Tables, Views, and Aliases ---- Row 1 to 4 of 4
Command ===> BR                                              Scroll ===> CSR

Commands: GRANT  MIG
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
 V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
 ? - Show all line commands

Sel   Name                Owner    T DB Name  TS Name    Cols        Rows Checks
      *                   *        * *        *            *           *      *
----- ------------------ -------- - -------- -------- ------ ----------- ------
      GLWTDNG             DSG24D03 T DSG24D03 GLWS002       2           0      0
      GLWTENG             DSG24D03 T DSG24D03 GLWS002       2           0      0
      GLWTPNG             DSG24D03 T DSG24D03 GLWS002       2           0      0
      GLWTLNG             DSG24D03 T DSG24D03 GLWS002       2           0      0
****************************** END OF DB2 DATA ******************************
```

Figure 3-22 Table display panel

The result is the browse panel shown in Figure 3-23. This panel can be scrolled in all directions.

```
DB2 Admin -- Browse Result of SQL Select       ------ Line 00000000 Col 001 080
Command ===>                                                 Scroll ===> CSR

********************************** Top of Data **********************************
NAME    CREATOR  TYPE DBNAME   TSNAME    DBID   OBID COLCOUNT EDPROC VALPROC CL
------- -------- ---- -------- ------- ------ ------ -------- ------ ------- --
GLWTDNG DSG24D03 T    DSG24D03 GLWS002    292     68        2
GLWTENG DSG24D03 T    DSG24D03 GLWS002    292     71        2
GLWTPNG DSG24D03 T    DSG24D03 GLWS002    292     74        2
GLWTLNG DSG24D03 T    DSG24D03 GLWS002    292     77        2
******************************* Bottom of Data ******************************
```

Figure 3-23 Showing all rows in BROWSE format

If you only want to see *all columns of a single row*, you can use another option, the **I** line command. Entering the **I** command next to the first row of Figure 3-22 displays all columns for that row, as shown in Figure 3-24.

```
DB2 Admin ------------------- DB8A Display Row ------------- Row 1 to 36 of 52
Command ===>                                                  Scroll ===> CSR


S Column Name        Column Value
  *                  *
- ------------------ ------------------------------------------------------------
  NAME               GLWTDNG
  CREATOR            DSG24D03
  TYPE               T
  DBNAME             DSG24D03
  TSNAME             GLWS002
  DBID                  292
  OBID                   68
  COLCOUNT                2
  EDPROC
  VALPROC
  CLUSTERTYPE
  CLUSTERRID                    0
  CARD                          0
  NPAGES                        0
```

Figure 3-24 Show all columns of a single row using the / line command

3.2.2 Optimizing the SQL search

To ensure a good response time when you are searching for objects in the DB2 catalog, keep the following recommendations in mind to speed up response times, especially when you have a large DB2 installation.

DB2 Admin generates SQL SELECT statements based on the selection criteria you specify. For performance reasons and to get a better quality of your results you can optionally enter one or more selection criteria on the bottom part of the catalog search panel.

You can qualify your searches for database (D), collection (L) and schema objects (H) by entering a matching pattern in the *In D/L/H* input field in Figure 3-13 on page 55.

If you are running against the actual DB2 catalog tables, rather than using a copy, you should keep in mind that there are only a limited number of indexes on the DB2 catalog. Depending on the size of your catalog, the search might have to read a lot of pages. Table 3-1 shows the search options and the search criteria you can provide for better performance.

Table 3-1 Performance considerations

Search option	Searching for	Fill in at least one of these fields
P, L	Plans, Collections	Name
S	Table spaces	In DB
C, Y	Columns, Synonyms	Creator
T,V,A,X	Tables, Views, Aliases, Indexes	Creator, In DB
K	Packages	In Col
xA	Authorizations	Grantee, Grantor

Primary commands to optimize the SQL search

There are some other primary commands that can be used to add additional search functions when selecting objects from the DB2 system catalog:

xSEL To use the *DB2 Admin SQL Select Prototype* option on the catalog tables, enter the object code (x, like "S" for table space) followed by the characters SEL. DB2 Admin displays the result on the panel that it would normally use for that object. This, then, allows you to use line commands on the resulting table.

xC To further qualify your search, you can use additional selection criteria fields as you find on the bottom of panel ADB21 (see Figure 3-13 on page 55). The option code (x) for an object followed by a **C** shows the columns for this object type and allows you to specify additional search arguments based on the columns in the DB2 catalog for that object.

DDL Jump key to go to the create DDL panel (equivalent to option 2.4).

DCL Jump key to go to the grant/revoke panel (equivalent to option 2.5).

Using the xSEL option

You can invoke the DB2 Admin *SQL Select Prototype* option on the catalog tables from the System Catalog panel. To invoke this function, you specify the object code followed by the characters SEL. As a result, DB2 Admin displays a SELECT statement that it would normally use for that object. By modifying this statement, you can build your own query.

For example, if you want to see all table spaces with more than one partition, you can enter **SSEL** (S for table space) in the option field of System Catalog - object options panel (see Figure 3-10 on page 53). You get the Build SQL SELECT Prototype panel for the DB2 catalog table *SYSTABLESPACE*, as shown in Figure 3-25.

```
DB2 Admin --------- DB8A Build SQL SELECT Prototype -------- Row 1 to 28 of 41
Command ===>                                                 Scroll ===> CSR

SELECT T.*,NAME
  FROM SYSIBM.SYSTABLESPACE T
  WHERE PARTITIONS>1
ORDER BY NAME ASC
GROUP BY
Commands: EDIT  RESET  *  QUOTE  INS  UPD  DEL  COUNT  COUNT_BIG
Line commands:  S - Show  SA - Show ASC  SD - Show DESC
 AVG, COUNT, COUNT_BIG, MAX, MIN, STDDEV, SUM, VARIANCE - Aggregate functions
 <oper><expr>, OR <pred>, IN list, BETWEEN <expr>,<expr> - WHERE predicates
Select                            Column Name        Col Type      Length
                                  *                  *                  *
--------------------------------- ------------------ -------- -----------
*SA                               NAME               VARCHAR           24
                                  CREATOR            VARCHAR          128
                                  DBNAME             VARCHAR           24
                                  DBID               SMALLINT           2
                                  OBID               SMALLINT           2
                                  PSID               SMALLINT           2
                                  BPOOL              CHAR               8
*>1                               PARTITIONS         SMALLINT           2
                                  LOCKRULE           CHAR               1
```

Figure 3-25 Build SQL SELECT Prototype panel

There you can modify the SELECT statement by specifying a WHERE condition (**>1** for the PARTITION column) and an ascending ORDER BY (**SA** clause for the NAME column).

You can change your selections at any time by issuing the **RESET** command. After pressing the Enter key, DB2 Admin returns a list of all table spaces having more than one partition, as shown in Figure 3-26.

```
DB2 Admin ------------------ DB8A Table Spaces --------------- Row 1 to 9 of 9
Command ===>                                                  Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line commands

Select Name     DB Name   Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
       *        *             * *      * * * * *      *           *      * * *
------ -------- -------- ------ ------ - - - - - ------ ----------- ------ - -
       DSN8S81E DSN8D81A      5 BP0    P N A N N      1         120      0   Y
       GLWSEMP  DSG24D02      4 BP15   A N A N Y      1          -1      0   Y
       GLWSEMP  DSG24D01      4 BP15   A N A N Y      1          -1      0   Y
       GLWSEMP  DSG24D04      4 BP15   A N A N Y      1          -1      0   Y
       GLWSEMP  DSG24D03      4 BP15   A N A N Y      1         144      0   Y
       GLWSSPL  DSG24D02      4 BP15   A N A N Y      1          -1      0   Y
       GLWSSPL  DSG24D03      4 BP15   A N A N Y      1         432      0   Y
       GLWSSPL  DSG24D04      4 BP15   A N A N Y      1          -1      0   Y
       GLWSSPL  DSG24D01      4 BP15   A N A N Y      1          -1      0   Y
****************************** END OF DB2 DATA ******************************
```

Figure 3-26 List table spaces with more than one partition sorted by name

Using the xC command

If you want to use additional search criteria for objects, you can type in the option code for an object, followed by a “C”. This shows you the columns for that object and you can define additional search arguments.

Archived

For example, if you only want to see table spaces that have names that start with GLW and reside in Buffer Pool 1, you specify the fields shown in bold in Figure 3-27.

```
DB2 Admin --------------------- DB8A System Catalog --------------------- 14:09
Option ===> S


Object options:                                          DB2 System: DB8A
  AO - Authorization options                             DB2 SQL ID: PAOLOR3
   G - Storage groups                P - Plans
   D - Databases                     L - Collections
   S - Table spaces                  K - Packages
   T - Tables, views, and aliases    M - DBRMs
   V - Views                         H - Schemas
   A - Aliases                       E - User defined data types
   Y - Synonyms                      F - Functions
   X - Indexes                       O - Stored procedures
   C - Columns                       J - Triggers
   N - Constraints                   Q - Sequences
  DS - Database structures         DSP - DS with plans and packages

Enter standard selection criteria (Using a LIKE operator, criteria saved):
Name     ===> GLW                >  Grantor  ===>          >
Owner    ===>          >            Grantee  ===>          >
In D/L/H ===>                    >
And/or other selection criteria (option xC shows you columns for option x)
Column   ===> BPOOL              >  Operator ===> =      Value ===> BP1
```

Figure 3-27 Select table spaces in BP1 that begin with GLW

Figure 3-28 shows you the result of all the qualifying table spaces.

```
DB2 Admin ------------------ DB8A Table Spaces --------------- Row 1 to 9 of 9
Command ===>                                                  Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line commands

Select Name     DB Name   Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
       *        *            * *      * * * * *      *           *      * * *
------ -------- -------- ------ ------ - - - - - ------ ----------- ------ - -
       GLWS001  GLWSAMP       0 BP1    A N A N Y      7          48      4   Y
       GLWS002  GLWSAMP       0 BP1    A N A N Y      4          36      4   Y
       GLWS003  GLWSAMP       0 BP1    A N A N Y     11         540     32   Y
       GLWSDPT  GLWSAMP       0 BP1    A N A N Y      1          48     32   Y
       GLWSEMP  GLWSAMP       4 BP1    A N A N Y      1        1440      0   Y
       GLWSEPA  GLWSAMP       0 BP1    A N A N Y      1         720     64   Y
       GLWSPJA  GLWSAMP       0 BP1    A N A N Y      1         360     32   Y
       GLWSPRJ  GLWSAMP       0 BP1    A N A N Y      1         360     32   Y
       GLWSSPL  GLWSAMP       4 BP1    A N A N Y      1         480      0   Y
******************************** END OF DB2 DATA ********************************
```

Figure 3-28 Resulting list of table spaces with a name starting with GLW in BP1

If you do not know the column name of the DB2 system catalog table you want to specify as additional selection criteria, you can use the **xC** command. In our case we use **SC**, because we are interested in the columns of the SYSTABLESPACE catalog table.

This gives you a list with all column names of that DB2 catalog table, and you can select the ones to fill in the selection criteria. The result of the **SC** command is shown in Figure 3-29.

```
DB2 Admin ----------- DB8A Columns in Table: SYSIBM.SYSTABLE Row 1 to 34 of 41
Command ===>                                                  Scroll ===> CSR

Line commands:
 T - Tables  X - Indexes  A - Auth  GR - Grant  H - Homonyms

Select Column Name        Col No Col Type Length      Scale  Null
       *                     * *                  *      * *
------ ------------------ ------ -------- ----------- ------ ----
       NAME                    1 VARCHAR          24      0 N
       CREATOR                 2 VARCHAR         128      0 N
       DBNAME                  3 VARCHAR          24      0 N
       DBID                    4 SMALLINT          2      0 N
       OBID                    5 SMALLINT          2      0 N
       PSID                    6 SMALLINT          2      0 N
       BPOOL                   7 CHAR              8      0 N
       PARTITIONS              8 SMALLINT          2      0 N
       LOCKRULE                9 CHAR              1      0 N
       PGSIZE                 10 SMALLINT          2      0 N
       ERASERULE              11 CHAR              1      0 N
       STATUS                 12 CHAR              1      0 N
       IMPLICIT               13 CHAR              1      0 N
       NTABLES                14 SMALLINT          2      0 N
       NACTIVE                15 INTEGER           4      0 N
```

Figure 3-29 Column names of SYSIBM.SYSTABLESPACE

Archived

3.2.3 Printing output to the print data set

If you want to write the output of a query to a data set, this can be done by using the print data set.

To use the print data set, we first have to build an SQL Prototype. Selecting option **2** from the DB2 Administration Menu (see Figure 3-3 on page 47) brings us to the Execute SQL Statements panel of Figure 3-30.

```
DB2 Admin -------------------- Execute SQL Statements ------------------- 14:55
Option ===> 3

   1 - Execute SQL statements from screen input          DB2 System: DB8A
   2 - Run or Explain SQL statements                     DB2 SQL ID: PAOLOR3
   3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
   4 - Create/drop/label/comment on objects
   5 - Grant/revoke privileges on objects
```

Figure 3-30 Execute SQL Statements panel

Selecting option **3** takes us to Figure 3-31, where we now enter the GLWTEMP table.

```
DB2 Admin ------------- Build SQL Prototype: Search Objects ------------- 15:21
Command ===>

Enter/verify:
 Owner    ===> DSG24D03 >           (optional, default is PAOLOR3)
 Name     ===> GLWTEMP            >  (optional)
```

Figure 3-31 Build SQL Prototype panel

This brings us to the Build SQL Prototype options panel as shown in Figure 3-32.

```
DB2 Admin ----------- DB8A Tables, Views, and Aliases -------- Row 1 to 1 of 1
Command ===>                                              Scroll ===> CSR

Line commands:
 SEL - Select for SQL SELECT prototype  T - Table
 DEL - DELETE prototype  INS - INSERT prototype  UPD - UPDATE prototype

Select  Name              Owner    T
        *                 *        *
------- ----------------- -------- -
S       GLWTEMP           DSG24D03 T
****************************** END OF DB2 DATA ******************************
```

Figure 3-32 Build SQL prototype options panel

Archived

We specify **S** for SELECT on the GLWTEMP table and we get the SQL statement we want to run as shown in Figure 3-33.

```
DB2 Admin --------- DB8A Build SQL SELECT Prototype -------- Row 1 to 19 of 19
Command ===>                                                  Scroll ===> CSR

SELECT EMP_NO,FIRSTNME,MIDINIT,LASTNAME,WORKDEPT,PHONENO,HIREDATE,JOB,MANAG >
  FROM DSG24D03.GLWTEMP T
  WHERE EMP_NO>1
ORDER BY EMP_NO ASC
GROUP BY ?
Commands: EDIT  RESET  *  QUOTE  INS  UPD  DEL  COUNT  COUNT_BIG
Line commands:  S - Show  SA - Show ASC  SD - Show DESC
 AVG, COUNT, COUNT_BIG, MAX, MIN, STDDEV, SUM, VARIANCE - Aggregate functions
 <oper><expr>, OR <pred>, IN list, BETWEEN <expr>,<expr> - WHERE predicates
Select                              Column Name        Col Type      Length
                                    *                  *                  *
---------------------------------- ------------------ -------- -----------
*SA                                 EMP_NO             INTEGER            4
*S                                  FIRSTNME           VARCHAR           12
*S                                  MIDINIT            CHAR               1
*S                                  LASTNAME           VARCHAR           20
*S                                  WORKDEPT           INTEGER            4
*S                                  PHONENO            CHAR               4
*S                                  HIREDATE           DATE              10
*S                                  JOB                CHAR              12
*S                                  MANAGER            CHAR               1
*S                                  EDLEVEL            SMALLINT           2
*S                                  SEX                CHAR               1
*S                                  BIRTHDATE          DATE              10
*S                                  SALARY             DECIMAL            9
*S                                  BONUS              DECIMAL            9
*S                                  COMM               DECIMAL            9
*S                                  CREATED_TS         TIMESTMP          26
*S                                  CREATED_BY         CHAR               8
*S                                  UPDATED_TS         TIMESTMP          26
*S                                  UPDATED_BY         CHAR               8
****************************** END OF DB2 DATA ******************************
```

Figure 3-33 Build SQL Select Prototype panel

Executing the statement results in the table display panel of Figure 3-34. As shown in “Using the BROWSE command” on page 62, if you want to see all columns of the results table, you can issue the **browse** command.

However, you can also use the **>PRINT TABLE** command to write the results to the print data set which we created in “Allocating the print data set” on page 49.

Issuing this command takes you to the print layout panel of Figure 3-35. This panel allows you to select the columns you want to write to the print data set by placing an **S** beside them.

```
DB2 Admin -- Browse Result of SQL Select                -- Row 1 to 37 of 1,000
Command ===> >PRINT TABLE                                        Scroll ===> CSR

L      EMP_NO FIRSTNME     MIDINIT LASTNAME            WORKDEPT PHONENO
            * *            *       *                          * *
- ----------- ------------ ------- ---------------- ----------- -------
            2 Jeremy       S       Ellis                     60 8813
            3 Lenny        M       Raue                      10 5782
            4 Tony         J       Matthews                  60 3589
            5 Martin               Mcconnel                  60 1050
            6 Stephen      L       Latham                     6 2885
            7 Martin       D       Sturges                   60 8638
            8 Dave         P       Beardsley                135 6716
            9 Adam         M       Robson                    80 7250
           10 Gary         T       Clark                    135 8030
           11 Rachel       P       Swanton                  112 7318
           12 Larry        L       Daraz                     60 3285
           13 Micheila     L       Jones                     24 5735
           15 Frank                Toovey                   151 8209
           16 Catherine    I       Patel                    135 9464
           17 Fiona        M       Pfann                     39 5524
           19 Valerie      R       Thomas                    10 1215
```

Figure 3-34 Issuing the >PRINT TABLE command

```
DB2 Admin ------------------ DB8A Print Layout ------------- Row 1 to 20 of 20
Command ===>                                                      Scroll ===> CSR

Current print columns:


Select Column Name        Col No Col Type Length Scale
       *                     * *            *      *
------ ------------------ ------ -------- ------ ------
S      EMP_NO                  1 INTEGER       4      0
S      FIRSTNME                2 VARCHAR      12      0
       MIDINIT                 3 CHAR          1      0
S      LASTNAME                4 VARCHAR      20      0
S      WORKDEPT                5 INTEGER       4      0
       PHONENO                 6 CHAR          4      0
       HIREDATE                7 DATE         10      0
S      JOB                     8 CHAR         12      0
       MANAGER                 9 CHAR          1      0
       EDLEVEL                10 SMALLINT      2      0
       SEX                    11 CHAR          1      0
       BIRTHDATE              12 DATE         10      0
       SALARY                 13 DECIMAL       9      2
       BONUS                  14 DECIMAL       9      2
       COMM                   15 DECIMAL       9      2
       CREATED_TS             16 TIMESTMP     26      0
       CREATED_BY             17 CHAR          8      0
       UPDATED_TS             18 TIMESTMP     26      0
       UPDATED_BY             19 CHAR          8      0
       EMP_NO                 20 INTEGER       4      0
****************************** END OF DB2 DATA ******************************
```

Figure 3-35 Selecting columns for the print data set

Archived

We can then view or edit this output in our print data set using the ISPF options as shown in Figure 3-36.

```
VIEW       PAOLOR3.PRINT                                    Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001      EMP_NO FIRSTNME     LASTNAME                WORKDEPT JOB
000002 ----------- ------------ -------------------- ----------- ------------
000003           2 Jeremy       Ellis                        60 CLERK
000004           3 Lenny        Raue                         10 MARKETING
000005           4 Tony         Matthews                     60 ADVERTISING
000006           5 Martin       Mcconnel                     60 MARKETING
000007           6 Stephen      Latham                        6 ARCHITECT
000008           7 Martin       Sturges                      60 ENGINEER
000009           8 Dave         Beardsley                   135 ADVERTISING
000010           9 Adam         Robson                       80 OPERATOR
000011          10 Gary         Clark                       135 SYSTEMS ENG
000012          11 Rachel       Swanton                     112 TEACHER
000013          12 Larry        Daraz                        60 ANALYST
000014          13 Micheila     Jones                        24 CLERK
000015          15 Frank        Toovey                      151 CLERK
000016          16 Catherine    Patel                       135 CARPENTER
000017          17 Fiona        Pfann                        39 ARTIST
000018          19 Valerie      Thomas                       10 MARKETING
```

Figure 3-36 Contents of print data set

3.3 Creating new objects based on existing objects

In this section we describe how you can use DB2 Admin to create new DB2 objects by using existing definitions in the DB2 catalog.

Archived

3.3.1 Creating a limited number of objects (DDL)

Using the **DDL** line command, you can generate the DDL for a single object only. This function can be used for modeling DDL statements. For example, if you want the DDL for a certain table, you type in the **DDL** line command in front of that table (see Figure 3-37).

```
 DB2 Admin --------------- DB8A Tables, Views, and Aliases ---- Row 1 to 8 of 8
 Command ===>                                                 Scroll ===> CSR

 Commands: GRANT  MIG
 Line commands:
  C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
  V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
  ? - Show all line commands

 Sel   Name                Owner    T DB Name  TS Name    Cols        Rows Checks
       *                   *        * *        *             *           *      *
 ----- ------------------ -------- - -------- -------- ------ ----------- ------
       GLWTDPT             DSG24D03 T DSG24D03 GLWSDPT      10         132      0
DDL    GLWTEMP             DSG24D03 T DSG24D03 GLWSEMP      19        1660      0
       GLWTEPA             DSG24D03 T DSG24D03 GLWSEPA      10       13081      0
       GLWTPJA             DSG24D03 T DSG24D03 GLWSPJA      10        5148      0
       GLWTPRJ             DSG24D03 T DSG24D03 GLWSPRJ      13         286      0
       GLWTSPL             DSG24D03 T DSG24D03 GLWSSPL       6        3342      0
       GLWTACT             DSG24D03 T DSG24D03 GLWS001       3          18      0
       GLWTJBS             DSG24D03 T DSG24D03 GLWS001       4          15      0
 ****************************** END OF DB2 DATA ******************************
```

Figure 3-37 Generating DDL for a table

The result is a temporary file with a CREATE statement (see Figure 3-38) which can be modified, stored, or executed as described below.

Archived

```
EDIT        SYS07078.T142115.RA000.PAOLOR3.R0110352         Columns 00001 00072
 Command ===>                                                  Scroll ===> CSR
 ****** ***************************** Top of Data ******************************
 000001   SET CURRENT SQLID='DSG24D03';
 000002   CREATE TABLE DSG24D03.GLWTEMP
 000003      (EMP_NO               INTEGER NOT NULL,
 000004       FIRSTNME             VARCHAR(12) FOR SBCS DATA NOT NULL,
 000005       MIDINIT              CHAR(1) FOR SBCS DATA NOT NULL,
 000006       LASTNAME             VARCHAR(25) FOR SBCS DATA NOT NULL,
 000007       WORKDEPT             INTEGER WITH DEFAULT NULL,
 000008       PHONENO              CHAR(4) FOR SBCS DATA NOT NULL,
 000009       HIREDATE             DATE NOT NULL WITH DEFAULT,
 000010       JOB                  CHAR(12) FOR SBCS DATA WITH DEFAULT NULL,
 000011       MANAGER              CHAR(1) FOR SBCS DATA NOT NULL,
 000012       EDLEVEL              SMALLINT NOT NULL,
 000013       SEX                  CHAR(1) FOR SBCS DATA WITH DEFAULT NULL,
 000014       BIRTHDATE            DATE NOT NULL WITH DEFAULT,
 000015       SALARY               DECIMAL(9, 2) WITH DEFAULT NULL,
 000016       BONUS                DECIMAL(9, 2) WITH DEFAULT NULL,
 000017       COMM                 DECIMAL(9, 2) WITH DEFAULT NULL,
 000018       CREATED_TS           TIMESTAMP NOT NULL WITH DEFAULT,
 000019       CREATED_BY           CHAR(8) FOR SBCS DATA NOT NULL
 000020         WITH DEFAULT,
 000021       UPDATED_TS           TIMESTAMP NOT NULL WITH DEFAULT,
 000022       UPDATED_BY           CHAR(8) FOR SBCS DATA NOT NULL
 000023         WITH DEFAULT,
 000024       CONSTRAINT GLWPEMP
 000025       PRIMARY KEY (EMP_NO))
 000026     IN DSG24D03.GLWSEMP
 000027     AUDIT NONE
 000028     DATA CAPTURE NONE
 000029     CCSID      EBCDIC
 000030     NOT VOLATILE;
 000031   COMMIT;
 ****** **************************** Bottom of Data ****************************
```

Figure 3-38 Generated DDL for a table

After pressing PF3, you are asked whether you want to execute this DDL or store the statement in a data set for later use, as shown in Figure 3-39.

```
 DB2 Admin ----------------- DB8A SQL - Execute & Copy ----------------- 14:29
 Option ===>
 SQL has been copied.
    Execute the generated SQL : YES     (Yes/No)

    Copy the SQL to dataset . : 'ADMIN.DDL.DATA(TABLE)'                      >
      Disposition . . . . . . : SHR      (SHR, OLD or MOD)
      The specified dataset will be created if it does not exist.
```

Figure 3-39 SQL - Execute & Copy panel

In this example, we decide to store the DDL for later use. After pressing Enter, we get notified that our SQL got stored in the specific data set ADMIN.DDL.DATA(TABLE) as shown in Figure 3-39.

After the message `SQL has been copied` appears, we press PF3 to get to the execution prompt. We then specify to add the DDL statement to a Work Statement List (WSL) with option **4** as shown in Figure 3-40.

```
DB2 Admin --------------- DB8A Statement Execution Prompt --------------- 21:44
Option ===> 4

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now (add an A for all stmts. For example 1A - Execute all stmts):
   1 - Execute the statement
   2 - Edit the statement
   3 - Create a batch job with the statement
   4 - Add the statement to the work statement list
 CAN - Cancel
 Work statement list dsn  ===> 'ADMIN.CM.WSL'
 Work statement list name ===>          Action ===> A (Append or Replace)
                                                          More:     +
Statement that is about to be executed (first 28 lines, more stmts pending):
  SET CURRENT SQLID='DSG24D03'
```

Figure 3-40 Statement Execution Prompt panel

After pressing Enter, the DDL statement is added to the WSL. If we then leave the panel with PF3, the CM Prompt pop-up panel appears (because userid PAOLOR3 is defined as an optional user of CM) if we wish to use Change Management for this statement, as shown in Figure 3-41.

```
DB2 Admin -------------- DB8A Generate SQL from DB2 catalog ------------- 21:44
O EsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
  e DB2 Admin ----------- DB8A Change Management Prompt ----------- 21:56 e
  e                                                                      e
  e Change Management is optional for SQLID: PAOLOR3                     e
  e                                                                      e
  e Do you wish to use Change Management for the statement: NO  (Yes/No)  e
  e                                                                      e
  e                                                        More:     +   e
  e Statement that is about to be executed (first 28 lines)              e
  e   CREATE TABLE DSG24D03.GLWTEMP                                      e
  e      (EMP_NO              INTEGER NOT NULL,                          e
  e       FIRSTNME            VARCHAR(12) FOR SBCS DATA NOT NULL,        e
  e       MIDINIT             CHAR(1) FOR SBCS DATA NOT NULL,            e
  e       LASTNAME            VARCHAR(25) FOR SBCS DATA NOT NULL,        e
  e       WORKDEPT            INTEGER WITH DEFAULT NULL,                 e
  DssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

Figure 3-41 Change Management Prompt panel

We specify **NO** and return to the originating panel (ADB21T) as shown in Figure 3-37 on page 73, where we issued the **DDL** line command initially.

3.3.2 Creating many objects including dependencies (GEN)

DB2 Admin has a reverse engineering function (GEN) that lets you create objects and their dependencies from the DB2 system catalog.

You can, for example, create DB2 “Model” objects which meet your company’s standards. If you want to create a new object, you can use the Model object as a basis to generate new objects. You can use masking with the GEN function, which enables you to easily change the model names into the real names that are required for the objects you want to create.

We use the GEN function to create a database for a new project. The model for our generation is the development database “DSG24D03”. First, look up this database in the DB2 system catalog and enter the **GEN** line command, as shown in Figure 3-42.

```
DB2 Admin ------------------- DB8A Databases ---------------- Row 1 to 1 of 1
Command ===>                                                 Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                        Storage  Buffer          Created      Index
Select Name     Owner    Group    Pool       DBID By       T E Buffer Pool
       *        *        *        *            * *        * * *
------ -------- -------- -------- -------- ------ -------- - - -----------
GEN    DSG24D03 PAOLOR2  GLWG01   BP15        292 PAOLOR2    E BP16
****************************** END OF DB2 DATA ******************************
```

Figure 3-42 Generating a database

Archived

This takes you to the Generate SQL from DB2 catalog panel (see Figure 3-43).

```
DB2 Admin -------------- DB8A Generate SQL from DB2 catalog ------------- 14:50
Option ===>

 Generate SQL statements for database DSG24D03          DB2 System: DB8A
                                                        DB2 SQL ID: PAOLOR3

 SQL statement types to be generated from the DB2 catalog:
   CREATE DATABASE. . . . . : Y (Y,N)   GRANT access ON DATABASE . : Y (Y,N)
   CREATE TABLESPACE. . . . : Y (Y,N)   GRANT access ON TABLESPACE : Y (Y,N)
   CREATE TABLE . . . . . . : Y (Y,N)   GRANT access ON TABLE. . . : Y (Y,N)
   CREATE VIEW. . . . . . . : D (Y,N,D) GRANT access ON VIEW . . . : N (Y,N)
   CREATE INDEX . . . . . . : Y (Y,N)   ALTER TABLE ADD FOREIGN KEY: D (Y,N,D)
   CREATE SYNONYM . . . . . : N (Y,N)   LABEL ON . . . . . . . . . : N (Y,N)
   CREATE ALIAS . . . . . . : N (Y,N)   COMMENT ON . . . . . . . . : N (Y,N)
   CREATE TRIGGER . . . . . : Y (Y,N,D) REBIND PLAN/PACKAGE. . . . : N (Y,N)
   CREATE STORAGE GROUP . . : N (Y,N)   GRANT use OF STORAGE GROUP : N (Y,N)

 New names/values for generated SQL: (leave blank to use current values)
   Object owner. . . . . : SG24D03P >  Run SQLID. . . . . . . . . : SG24D03P
   Object grantor. . . . :          >
   Alloc TS size as. . . : DEFINED     (DEFINED, USED, or ALLOC)
   Database name . . . . : DSG24P03
   Storage group for TS. : GLWG01   >  Storage group for IX . . . : GLWG01   >
   Target DB2 version. . :             (Current DB2 version: 815)
   Use Masking . . . . . : NO          (Yes/No)
   Generate catalog stats: NO          (Yes/No/Only)
     Target cat qualifier:          >  (Default is SYSIBM)

 Output file and execution mode:
   Add to work stmt list : NO          (Yes/No)
   Data set name . . . . : 'PAOLOR3.SPUFI.DATA(DSG24P03)'                   >
     Data set disposition: OLD         (OLD, SHR, or MOD)
   Execution mode. . . . : BATCH       (BATCH or TSO)
   Commit statements per :             (Db, tS, Tb, All, None. Default is All)
   DB2 defaults handling :             (Keep, or Remove. Default is Keep)

 DB2 Command output file:
   Data set name . . . . :                                                  >
     Data set disposition: OLD         (OLD, SHR, or MOD)

 BP - Change batch job parameters
```

Figure 3-43 Generating SQL for a database from DB2 catalog

On this panel we specify a **Y** for all objects we want to create using DB2 Admin.

For CREATE VIEW, we use **D**, which indicates that views are extracted, but DB2 Admin does not check that all other objects referenced in the view, also are being generated. This significantly reduces the resource consumption to generate all DDL statements, especially when running against a large DB2 catalog.

For ALTER TABLE ADD FOREIGN KEY, we entered **D** to extract FOREIGN KEYS for tables that are dependent on the tables being extracted.

We also define the names that should be changed and choose the following values for other fields:

Object owner	Our new owner is **SG24D03P**.
Run SQLID	**SG24D03P** should be used when creating objects.
Alloc TS size as	We want to use the **DEFINED** size for the table spaces.
Database name	Our new database name is **DSG24P03**.
Storage group	For both table spaces and index spaces, **GLWG01** should be used as the storage group name.
Use Masking	We specify **NO** because we do not want to change other names.
Add to work stmt list	We specify **NO** because do not want to save the output in a work statement list.
Data set name	We want to send the SQL output to **PAOLOR3.SPUFI.DATA(DSG24P03)**. If you do not specify a data set name, the output is routed to SYSOUT=*
Execution mode	As we prefer to run jobs in the background, **BATCH** is specified to generate a batch job for execution.
Commit statements per	We use the default (All).
DB2 defaults handling	We use the default (Keep).
Data set name	We do not specify a command output data set, since we do not ask for REBIND statements to be generated.

After pressing the Enter key, a job, executing program “ADB2GEN” is created. We submit this job and it creates the output in our defined data set member “PAOLOR3.SPUFI.DATA(DSG24P03)”.

We used this member to create the new database DSG24P03. The result, shown in Figure 3-44, is produced using the **DS** (Display Structure) line command.

Archived

```
DB2 Admin --------------- DB8A Database Structures --------- Row 1 to 34 of 75
Command ===>                                                   Scroll ===> CSR

Line commands:  S - Show object  DSN - Data sets
                                                        PSID/
Sel Type   Object Name              Qualifier  DBID ISOBID   OBID Note
    *      *                        *             *      *      * *
--- ------ -----------------------> -------- ------ ------ ------ -------------
    D----- DSG24P03----------------              310      0      0
     S       GLWS001                DSG24P03    310     44     43 Segmented
      T        GLWTACT              DSG24P03    310      0     45
       UC        GLWPACT                          0      0      0 Primary key
       X         GLWXACT1           DSG24P03    310     47     46
       V         WLACT              DSG24P03      0      0      0
      T        GLWTJBS              DSG24P03    310      0     48
       X         GLWXJBS1           DSG24P03    310     50     49
       V         WLJOBSALARY        DSG24P03      0      0      0
      T        GLWTLCN              DSG24P03    310      0     51
       X         GLWXLCN1           DSG24P03    310     53     52
       V         WLLOCN             DSG24P03      0      0      0
      T        GLWTLNM              DSG24P03    310      0     54
       X         GLWXLNM1           DSG24P03    310     56     55
       V         WLLASTNAME         DSG24P03      0      0      0
      T        GLWTPGW              DSG24P03    310      0     57
       X         GLWXPGW1           DSG24P03    310     59     58
       V         WLPROGWT           DSG24P03      0      0      0
      T        GLWTSFN              DSG24P03    310      0     60
       X         GLWXSFN1           DSG24P03    310     62     61
       V         WLSEXFIRSTNAME     DSG24P03      0      0      0
      T        GLWTVRN              DSG24P03    310      0     63
       X         GLWXVRN1           DSG24P03    310     65     64
       V         WLVERSN            DSG24P03      0      0      0
     S       GLWS002                DSG24P03    310     67     66 Segmented
```

Figure 3-44 Structure of new database DSG24P03

Note that a similar “tree structure,” with additional information on authorizations and RI relationships, can be obtained using drop impact analysis, as shown in Figure 3-59 on page 88.

3.4 Working with a copy of the DB2 system catalog

Many installations do not allow users to run queries against the actual DB2 catalog, especially on production systems. Instead, they use one or multiple copies of the actual DB2 system catalog. Using multiple copies of your local DB2 system catalog enables you to work with different versions of a DB2 system catalog, for example, to regenerate the DDL of an object that was dropped by accident on the real system. Another reason can be that you do not want to impact performance on your production DB2 system by running complex catalog searches.

How you create a catalog copy is described in Chapter 2. “Activating DB2 Admin” in *DB2 Administration Tool for z/OS User’s Guide and Reference Version 7 Release 2*, SC18-9552.

Archived

After enabling the use of catalog copies, you see an additional Switch Catalog Copy option on the System Catalog panel as shown in Figure 3-45.

```
DB2 Admin --------------------- DB8A System Catalog --------------------- 17:39
Option ===>


Object options:                                          DB2 System: DB8A
  AO - Authorization options                             DB2 SQL ID: PAOLOR3
   G - Storage groups                 P - Plans
   D - Databases                      L - Collections
   S - Table spaces                   K - Packages
   T - Tables, views, and aliases     M - DBRMs
   V - Views                          H - Schemas
   A - Aliases                        E - User defined data types
   Y - Synonyms                       F - Functions
   X - Indexes                        O - Stored procedures
   C - Columns                        J - Triggers
   N - Constraints                    Q - Sequences
  DS - Database structures          DSP - DS with plans and packages

Enter standard selection criteria (Using a LIKE operator, criteria saved):
Name     ===>                    >  Grantor  ===>          >
Owner    ===>          >            Grantee  ===>          >
In D/L/H ===>                    >  Switch Catalog Copy  ===> C   (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
Column   ===>                    >  Operator ===>        Value ===>
```

Figure 3-45 Switch Catalog Copy option

To switch to a copy of a DB2 system catalog, you enter **C** in the Switch Catalog Copy field. This gives you a list of existing catalog copies.

We customized our DB2 Admin system to allow working with a copy of the DB2 system catalog. We created a copy of the DB2 system catalog of DB2 system *DB8A*. By specifying option **C**, we get the panel shown in Figure 3-46.

```
DB2 Admin -------------- DB8A Select Copy of DB2 Catalog ----- Row 1 to 1 of 1
Command ===>                                              Scroll ===> CSR

DB2 Catalog Copy Version Selection:                      DB2 System: DB8A
                                                         DB2 SQL ID: PAOLOR3
 S - Select an entry
                                  Copy     Planname
Select Timestamp                  Owner    Suffix   Type Location
       *                          *        *        *    *
------ -------------------------- -------- -------- ---- ----------------
S      2007-02-28-18.22.51.263055 A72      CC       C
****************************** END OF DB2 DATA ******************************
```

Figure 3-46 Select Copy of DB2 Catalog panel

To switch to the catalog copy **A72**, just select this row. The System Catalog panel of Figure 3-47 shows that we are now working with a copy of a system catalog.

```
DB2 Admin -------------------- CCCC System Catalog -------------------- 18:28
Option ===>
DB2 Catalog switched

Owner of DB2 Catalog Tables : A72                        DB2 System: DB8A
  AO - Authorization options                             DB2 SQL ID: PAOLOR3
   G - Storage groups                P - Plans
   D - Databases                     L - Collections
   S - Table spaces                  K - Packages
   T - Tables, views, and aliases    M - DBRMs
   V - Views                         H - Schemas
   A - Aliases                       E - User defined data types
   Y - Synonyms                      F - Functions
   X - Indexes                       O - Stored procedures
   C - Columns                       J - Triggers
   N - Constraints                   Q - Sequences
  DS - Database structures         DSP - DS with plans and packages

Enter standard selection criteria (Using a LIKE operator, criteria saved):
Name     ===>                    >  Grantor  ===>         >
Owner    ===>          >            Grantee  ===>         >
In D/L/H ===>                    >  Switch Catalog Copy  ===> N   (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
Column   ===>                    >  Operator ===>        Value ===>
```

Figure 3-47 System Catalog panel after switching to a catalog copy

To switch back to the original catalog of DB2 system “DB8A”, enter **S** in the Switch Catalog Copy field.

3.5 Showing database structure and data set information

The option **DSP** on the System Catalog panel provides a structured list of the object that you selected with plans and packages and the structure of other objects within the database that are related to the object that you selected (see Figure 3-48).

```
DB2 Admin --------------- DB8A Database Structures -------- Row 1 to 35 of 141
Command ===>                                                 Scroll ===> CSR

Line commands:  S - Show object  DSN - Data sets
                                                        PSID/
Sel Type   Object Name              Qualifier  DBID ISOBID   OBID Note
    *      *                        *             *      *      * *
--- ------ -----------------------> -------- ------ ------ ------ -------------
    D----- DSG24D03----------------              292      0      0
     S       GLWS001                DSG24D03    292     44     43 Segmented
      T        GLWTACT              DSG24D03    292      0     45
       UC        GLWPACT                          0      0      0 Primary key
       X         GLWXACT1           DSG24D03    292     47     46
        K          ACTSEL           DSG24D03      0      0      0
       V         WLACT              DSG24D03      0      0      0
       K         PRJADD             DSG24D03      0      0      0
```

Figure 3-48 Object structure from database DSG24D03

An easy way to display all the associated data set names for a specific table space, index space, or partition, is to use the **DSN** line command (see Figure 3-49).

```
DB2 Admin --------------- DB8A Database Structures -------- Row 1 to 35 of 141
Command ===>                                                  Scroll ===> CSR

Line commands:  S - Show object  DSN - Data sets
                                                        PSID/
Sel Type   Object Name              Qualifier  DBID ISOBID   OBID Note
    *      *                        *             *      *      * *
--- ------ -----------------------> -------- ------ ------ ------ -------------
    D----- DSG24D03----------------              292      0      0
DSN  S       GLWS001                DSG24D03    292     44     43 Segmented
      T        GLWTACT              DSG24D03    292      0     45
       UC        GLWPACT                          0      0      0 Primary key
       X         GLWXACT1           DSG24D03    292     47     46
        K          ACTSEL           DSG24D03      0      0      0
       V         WLACT              DSG24D03      0      0      0
       K         PRJADD             DSG24D03      0      0      0
```

Figure 3-49 Display DSN that is associated with the table space, index space, or partition

The result is shown in Figure 3-50.

```
DB2 Admin -- DB8A Data Sets for TS DSG24D03.GLWS001    ------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR

Line commands:  LISTC - List catalog

Select Data Set Name
------ -------------------------------------------
LISTC  DB8AU.DSNDBC.DSG24D03.GLWS001.I0001.A001
****************************** END OF DB2 DATA ******************************
```

Figure 3-50 Data set name for table space DSG24D03.GLWS001

On the panel that contains the DSN information shown in Figure 3-50, you can also specify the **LISTC** line command. This command gives you VSAM cluster information for that table space or index space. See Figure 3-51 for the LISTCAT information of the data set.

```
 DB2 Admin ------------------ TSO LISTCAT Output Display ----------------- 20:42
 Command ===>                                                 Scroll ===> CSR
********************************** Top of Data ***********************************
IDCAMS  SYSTEM SERVICES                                          TIME: 20:42:14
CLUSTER ------- DB8AU.DSNDBC.DSG24D03.GLWS001.I0001.A001
     IN-CAT --- UCAT.SBOXB2
     HISTORY
       DATASET-OWNER-------BART     CREATION--------2007.053
       RELEASE----------------2     EXPIRATION------0000.000
       BWO STATUS--------(NULL)     BWO TIMESTAMP-----(NULL)
       BWO---------------(NULL)
     PROTECTION-PSWD-----(NULL)     RACF----------------(NO)
     ASSOCIATIONS
       DATA-----DB8AU.DSNDBD.DSG24D03.GLWS001.I0001.A001
   DATA ------- DB8AU.DSNDBD.DSG24D03.GLWS001.I0001.A001
     IN-CAT --- UCAT.SBOXB2
     HISTORY
       DATASET-OWNER-------BART     CREATION--------2007.053
       RELEASE----------------2     EXPIRATION------0000.000
       ACCOUNT-INFO-----------------------------------(NULL)
     PROTECTION-PSWD-----(NULL)     RACF----------------(NO)
```

Figure 3-51 VSAM cluster information for table space DSG24D03.GLWS001

3.6 Revoking impact

Before you execute a REVOKE SQL statement, you might want to see what implications this statement might have. *Revoke impact* is a preview function, before the REVOKE statement is actually executed. The revoke impact report is generated at your request.

As an example, we use this function for revoking some database authorizations from the userid “PAOLO”. In Figure 3-52 you see the list of all database authorizations that userid “PAOLO” has.

```
 DB2 Admin --------------- DB8A Database Authorizations ------- Row 1 to 2 of 2
 Command ===>                                                 Scroll ===> CSR

 Commands: REVOKE  GRANT
 Line commands:                                  C C D D D D D I L R R R S S S
  R - Revoke  GR - Grant  D - Database           R R B B B I R M O E E E T T T
  I - Interpretation                             E E A C M S O A A O C P A A O
                                                 T T D T A P P G D R O A R T P
                      Database Grant           H A S M R I D D E D G V I T S
 Sel Grantor  Grantee Name     Timestamp        G B     L N B B   B     R
     *        *       *        *                * * * * * * * * * * * * * * *
 --- -------- -------- -------- ---------------- - - - - - - - - - - - - - - -
     PAOLOR3  PAOLO    DSG24D02 2007-02-28-19.45 S Y Y       Y Y           Y   Y
 R   PAOLOR3  PAOLO    DSG24D03 2007-02-28-19.38 S G G G G G G G G G G G G G G
******************************** END OF DB2 DATA ********************************
```

Figure 3-52 Database authorizations for PAOLO on databases DSG24D0%

Entering the **R** line command gives you the list of privileges the grantee “PAOLO” has, as shown in Figure 3-53. If you want to keep some of the privileges, you can replace the **G** with a blank. If you enter **Y** in the Report Revoke Impacts field, you request revoke impact analysis.

```
DB2 Admin --------------- DB8A Revoke Database Privileges --------------- 19:52
Command ===>

 REVOKE                                                DB2 SQL ID: PAOLOR3

 Enter any character in front of the privilege to revoke it from the user:

 G DBADM         G CREATETAB     G DROP          G RECOVERDB     G STARTDB
 G DBCTRL        G CREATETS      G IMAGCOPY      G REORG         G STATS
 G DBMAINT       G DISPLAYDB     G LOAD          G REPAIR        G STOPDB

 ON DATABASE
   Database name ===> DSG24D03
 FROM
   From          ===> PAOLO    >
 BY
   By            ===>

 Report Revoke Impacts  ===> YES  (Yes/No)
 Report Dropped Synonyms & Aliases ===> NO   (Yes/No)
```

Figure 3-53 Revoke Database Privileges panel

The report is displayed as a tree structure. At the top of that structure, as shown in Figure 3-54, you see the authorization that is explicitly being revoked by the statement (level 0). Below that, you see a table privilege and an object, in our sample a view, that would be lost as a consequence of executing the REVOKE statement.

```
DB2 Admin -------------- DB8A Revoke Impact Report ----------- Row 1 to 3 of 3
Command ===>                                                 Scroll ===> PAGE

WARNING - You elected not to have dropped Synonyms and Aliases reported.
There may be some dropped Synonyms or Aliases that are not shown here.
Line commands:  I - Interpretation
                                     Owner/
S Cscd Grantee  Resource Name/     O Schema/  Grantor/ H Privileges/
  Lvl           K Collection Id    T P/K Name Binder   G Effect
- ---- -------- ------------------ - -------- -------- - -----------
  0    PAOLO    DSG24D03           D          PAOLOR3  S GGGGGGGGGGGGGGG
  1    DSN8810  GLWTDPT            T DSG24D03 PAOLO    D     Y
  1             VGLWTEMP           V PAOLO               DROP
******************************** END OF DB2 DATA ********************************
```

Figure 3-54 Revoke impact report for PAOLO on database DSG24D03

If you go back to the panel shown in Figure 3-53 and enter **NO** for “Report Revoke Impacts”, you go to the Statement Execution Prompt panel of Figure 3-55 (provided that prompting is active), which shows the generated REVOKE statement. You can then choose what to do with this statement.

```
DB2 Admin --------------- DB8A Statement Execution Prompt --------------- 20:06
Option ===>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
   1 - Execute the statement
   2 - Edit the statement
   3 - Create a batch job with the statement
   4 - Add the statement to the work statement list
 CAN - Cancel
 Work statement list dsn  ===> 'PAOLOR3.WORKLIST'
 Work statement list name ===> ALTERC    Action ===> A (Append or Replace)

Statement that is about to be executed (first 28 lines):
REVOKE CREATETAB,CREATETS,DBADM,DBCTRL,DBMAINT,DISPLAYDB,DROP,IMAGCOPY,L
OAD,REORG,RECOVERDB,REPAIR,STARTDB,STATS,STOPDB
  ON DATABASE DSG24D03
  FROM  "PAOLO"
```

Figure 3-55 Statement Execution Prompt panel

3.7 Dropping impact

DB2 Admin has the ability to provide *drop impact* reporting when dropping a DB2 object, for example, a table, view, or index. It is often important to show what other objects, plans, or packages are affected when you plan to drop a DB2 object.

This function analyzes the impact that the DROP of any DB2 object has on other DB2 objects in the DB2 subsystem.

Archived

To drop an object, you select it from the DB2 system catalog. Then you enter **DROP** next to the object you want to drop. In the example shown in Figure 3-56, we drop a table space from our database *DSG24D03*.

```
 DB2 Admin ------------------ DB8A Table Spaces ------------------ Row 1 from 8
 Command ===>                                                  Scroll ===> CSR

 Commands: GRANT  MIG  DIS  STA  STO
 Line commands:
  T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
  DIS - Display table space  STA - Start table space  STO - Stop table space
  ? - Show all line commands

 Select Name     DB Name   Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
        *        *             * *      * * * * *      *           *      * * *
 ------ -------- -------- ------ ------ - - - - - ------ ----------- ------ - -
        GLWSDPT  DSG24D03      0 BP15   A N A N Y      1         360     32   Y
        GLWSEMP  DSG24D03      4 BP15   A N A N Y      1         144      0   Y
        GLWSEPA  DSG24D03      0 BP15   A N A N Y      1         360     64   Y
        GLWSPJA  DSG24D03      0 BP15   A N A N Y      1         360     32   Y
DROP    GLWSPRJ  DSG24D03      0 BP15   A N A N Y      1         360     32   Y
        GLWSSPL  DSG24D03      4 BP15   A N A N Y      1         432      0   Y
        GLWS001  DSG24D03      0 BP15   A N A N Y      7          60      4   Y
        GLWS002  DSG24D03      0 BP15   A N A N Y      4         360      4   Y
******************************** END OF DB2 DATA ******************************
```

Figure 3-56 List of all table spaces in database DSG24D03

The panel in Figure 3-57 asks you whether or not you want to report the impacts of this drop operation (the default is **YES**). The two other choices you have are to create a batch job that produces the drop impact report (**Batch**) or to execute the drop statement directly (**NO**).

```
 DB2 Admin -------------------- DB8A Drop Table Space -------------------- 12:31
 Command ===>

  DROP TABLESPACE

  Database ===> DSG24D03 (? to look up)
  Name     ===> GLWSPRJ  (? to look up)

  Display Drop Impact Report ===> YES (Yes, No, or Batch)
```

Figure 3-57 Drop table space GLWSPRJ

As a result, you get the summary of the impact that the DROP TABLESPACE statement can have. The result for our statement is shown in Figure 3-58. The report has three categories that show the result of the analysis:

- The first category shows the items to DROP or REVOKE.
- The second category shows the items to invalidate.
- The third category shows the constraints to remove.

Each of these groups is broken into further details and shows the quantity.

```
DB2 Admin -------------- DB8A DROP Impact Analysis Summary -------------- 12:34
Command ===>                                                  Scroll ===> CSR

SQL Statement: DROP TABLESPACE    "DSG24D03"."GLWSPRJ"
Line commands:  S - Show   blank - Suppress

    Items to                  Items to          Constraints to
    DROP or REVOKE            Invalidate        Remove
    -------------- ----       -------- ----     ------------------------- ----
  S Databases . . :   0     S Aliases :   0     S Check Constraints. . . :   0
  S Table Spaces. :   1     S Packages:   5     S Referential Constraints:   0
  S Tables. . . . :   1     S Plans . :   0     S Unique Constraints . . :   1
  S Aux. Tables . :   0                ====                                 ====
  S Indexes . . . :   1                   5                                    1
  S Authorizations:   0
  S Synonyms. . . :   0
  S Views . . . . :   1
  S Procedures. . :   0
  S Functions . . :   0
  S Triggers. . . :   0
  S Dist. Types . :   0
  S Sequences . . :   0
                   ====
                      4
```

Figure 3-58 DROP impact analysis summary

In this case, the DROP TABLESPACE statement drops or revokes four items; five items are invalidated; and one constraint is removed.

If you require more detailed information, you can enter **S** next to an item. Then, after pressing **Enter**, DB2 Admin displays additional information for each selected category. Since S is the default, if you do not want this information for a particular category, you have to **blank** out the **S**. A sample report is shown in Figure 3-59.

In this detailed report you see the information about the catalog objects to be dropped and the packages that are invalidated as a consequence of executing the DROP.

Valid line commands are **S** to show additional details for the selected object, or **DRD** which is only valid for a row that represents a table that is defined with RESTRICT ON DROP.

You can sort this list by one or more columns or **RE-SORT** this list later in the original sort order.

If you issue the **DROP** command on this panel, the tool takes you to the Statement Execution Prompt panel again (if you have the Prompt option set to YES) where you can decide if and how to really execute the DROP statement.

```
DB2 Admin ---------- DB8A DROP Impact Analysis Details ----- Row 1 to 10 of 10
Command ===>                                                  Scroll ===> CSR

SQL Statement: DROP TABLESPACE    "DSG24D03"."GLWSPRJ"

Commands: RE-SORT  DROP
Line commands:  S - Show object  DRD - DROP RESTRICT on DROP

Sel Type   Object Name/Grantor>Grantee Owner    Note
    *      *                           *        *
--- ------ --------------------------- -------- ------------------------------
     S       DSG24D03.GLWSPRJ          PAOLOR2  Segmented
      T        GLWTPRJ                 DSG24D03
       UC        GLWPPRJ               PAOLOR2  Primary key
       X         GLWXPRJ1              DSG24D03
    V          WLPROJ                  DSG24D03 View of a Table
    K      DPTDEL,Collid=DSG24D03      PAOLOR2  Invalidated
    K      EMPDEL,Collid=DSG24D03      PAOLOR2  Invalidated
    K      PRJADD,Collid=DSG24D03      PAOLOR2  Invalidated
    K      PRJSEL,Collid=DSG24D03      PAOLOR2  Invalidated
    K      PRJUPD,Collid=DSG24D03      PAOLOR2  Invalidated
****************************** END OF DB2 DATA ******************************
```

Figure 3-59 Drop Impact analysis details panel

3.8 Restarting the monitor

When running the batch execution program ADBTEP2, a restart monitor registers the execution process. This program inserts rows into the checkpoint table with restart information in case the program terminates abnormally. These rows are identified by the user ID that runs the job and the worklist name that has been chosen during the definition phase of the batch execution (for example migrate).

How to create the checkpoint table is described in Chapter 2. “Activating DB2 Admin” in the *DB2 Administration Tool for z/OS User’s Guide and Reference Version 7 Release 2,* SC18-9552.

To see the contents of the checkpoint table, use option **Z** (DB2 system administration) from the DB2 Administration Menu, and then select **2B** (Display/manage batch checkpoint). This brings you to the panel shown in Figure 3-60.

Archived

```
DB2 Admin ------------ DB8A Manage Batch Job Checkpoint Table ----------- 13:25
Option ===> 1


Batch Job Checkpoint Table :  A72.ADBCHKPT              DB2 System: DB8A
                                                        DB2 SQL ID: PAOLOR3
   1 - Display Checkpoint Records
   2 - Display Checkpoint Table Status


 Enter Checkpoint Table Owner:

   Table Owner ===> A72

 Enter display selection criteria for option 1:

   Userid      ===>                  (default is '')

   Worklist    ===>                  (default is '')
```

Figure 3-60 Manage Batch Job Checkpoint Table panel

This panel shows you the name of the checkpoint table and allows you to display the contents of the checkpoint table. You can choose option **1** to display active checkpoint records. You can select a Userid and/or Worklist to reduce the result list.

The Display Batch Job Checkpoint Table panel in Figure 3-61 shows the rows in the table you have selected. In the records, you see from which statement and in which step a process can be restarted.

```
DB2 Admin ---------- DB8A Display Batch Job Checkpoint Table - Row 1 to 3 of 3
Command ===>                                                 Scroll ===> CSR

                                                        DB2 System: DB8A
                                                        DB2 SQL ID: PAOLOR3
Checkpoint Table:  A72.ADBCHKPT

Line commands:
 D - Delete/Terminate  I - Insert  U - Update  N - Skip-Next
                                             Commit      Restart     Restart
S Userid   Worklist Suffix   Time            Number      Command     Action
  *        *       *        *                         * *           *
- -------- -------- -------- ---------------> ----------- ----------- -------
  PAOLOR3  A0000005         2007-02-28-19.04          1 RUN            R
  PAOLOR3  A0000007         2007-02-28-19.36          5 RUN            R
  PAOLOR3  A0000008         2007-02-28-20.06          1 UNLOAD         C
****************************** END OF DB2 DATA ******************************
```

Figure 3-61 Display Batch Job Checkpoint Table

The line commands, listed at the top of the panel, can be used to change the content of the table.

You can for example, terminate, delete an active ADBTEP2 job, or update a record of an abnormally terminated job. You can also insert a new checkpoint record.

When a WSL abnormally terminates and an entry was made in the checkpoint table with the restart information, the userid field of the primary user cannot be overtyped on the *Update an Entry* panel.

If the WSL restart is initiated from another primary userid, the input stream (SQL statements, utilities, and DB2 commands) are restarted from the beginning and not at that point where it fails. This is because the WSL uses the userid of the submitter when it searches for an entry in the checkpoint table.

Attention: If you are restarting a WSL with another userid than the original submitter, you might lose data, because the WSL is processed from the beginning and hence might delete already unloaded data sets that are required in a later phase of the restart.

If you want to re-run the WSL from USERID-1 from its last successful instruction performed, the userid in the checkpoint table has to be changed from the original submitter (USERID-1) to the userid of the new submitter (USERID-2). See Example 3-1.

Example 3-1 Update userid in checkpoint table

```
UPDATE A72.ADBCHKPT
 SET USERID = 'PAOLOR2'
 WHERE USERID = 'PAOLOR3' AND SUFFIX = '' AND WORKLIST = 'A0000007'
```

Note: If you do not want the restart capability, you can modify your job card statement (see Figure 3-7 on page 50) and enter **N** in the “ADBTEP2 restart Y/N” field. In this case your jobs are not restarted. However, ADBTEP2 always takes checkpoints, so you can change from RESTART(NO) to RESTART(YES) after a failure if required.

Selecting option **2** on the Manage Batch Job Checkpoint Table panel, shown in Figure 3-60, displays the table information of the checkpoint table from the DB2 system catalog, as shown in Figure 3-62. With this option you can issue any request against the checkpoint table that is supported by DB2 Admin, such as GRANT and REVOKE.

```
DB2 Admin --------------- DB8A Tables, Views, and Aliases ---- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR

Commands: GRANT  MIG
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
 V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
 ? - Show all line commands

Sel   Name                Owner    T DB Name  TS Name    Cols        Rows Checks
      *                   *        * *        *             *           *      *
----- ------------------ -------- - -------- -------- ------ ----------- ------
      ADBCHKPT            A72      T A72DCH   ADBSCH       12           3      0
****************************** END OF DB2 DATA ******************************
```

Figure 3-62 Display Checkpoint Table

Attention: Check for the existence of restart records if you allow the re-use of the same worklist name in your environment. Running different scenarios with the same job and worklist name can lead to bad results if a restart record from a previously abnormally terminated job still exists without you being aware of it.

Note: The restart monitor uses the *worklist name* to identify the process that has to be restarted. This is not the same as the *work statement list name* (WSL) that is explained in the next section. However, when DB2 Admin generates ADBTEP2 jobs from a WSL, the first part is the WSL name.

3.9 Using work statement lists

Work statement lists (WSL) were introduced in Version 3 of DB2 Admin, and have since been enhanced in many areas. We give you a short overview of the functions and demonstrate how a WSL operates today.

3.9.1 Work statement list basics

A WSL is a collection of one or more sequential tasks that perform basic operations to implement DB2 database administration and DB2 change management tasks. Entries in a WSL can contain items in any of the following categories:

- DDL statements to CREATE, DROP, ALTER or RENAME objects
- DCL statements to GRANT or REVOKE privileges
- DML statements, such as INSERT, UPDATE, and DELETE
- DB2 commands, such as START, STOP, ALTER, and SET
- DSN commands that execute under the DSN command processor, such as BIND, REBIND, FREE, and RUN
- Utility statements
- DB2 Admin instructions

A WSL can be *cloned* to repeat changes made against an object on one system to any number of other systems that contains the same object. During the cloning operation, you are able to change names, buffer pools, storage groups, and sizes. When changes are executed, a set of jobs to implement the changes from the WSL is generated.

Archived

Figure 3-63 shows the basic functions of a WSL.

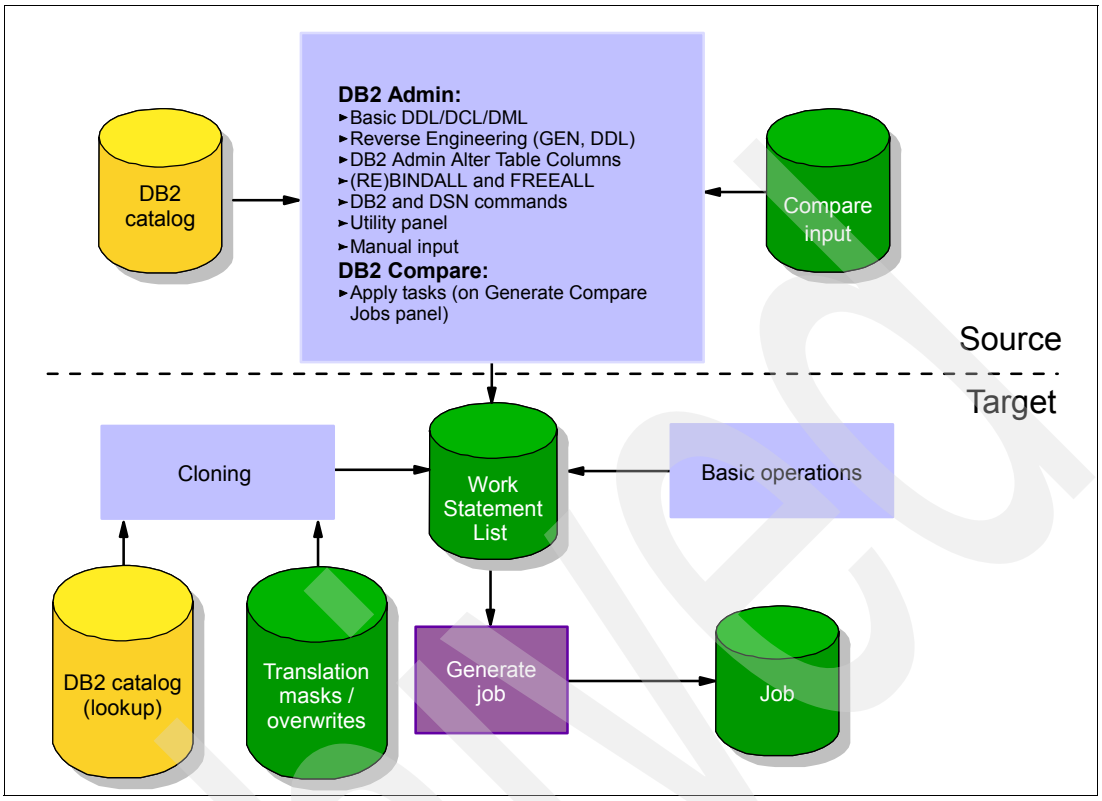

Figure 3-63 WSL operations

Note: DB2 Admin uses the term, *worklist*, when jobs for the migration and other functions are generated. The worklist name is being used as the middle qualifier for data sets that are created and used by these functions and for restart purposes. These worklists have nothing to do with work statement lists. To avoid any confusion, we use the abbreviation *WSL* when talking about work statement lists.

3.9.2 Creating a work statement list

You can create WSLs in several different ways, such as:

- By using DB2 Admin basic functions: definition SQL, authorization SQL, update SQL, DSN commands, and DB2 commands
- By using output from the DB2 Admin Reverse Engineering function
- By using the DB2 Admin Alter Table Columns function
- By using DB2 Compare apply tasks
- By using one of the DB2 Admin utilities panels
- By coding a WSL manually
- By cloning an existing WSL member

Depending on the settings of the prompt option (see 3.1.2, “Customizing your DB2 Admin parameters” on page 48) you are prompted before a statement or a command is executed. You can then choose to generate a WSL by selecting option **4** (Add the statement to the work statement list) on the Statement Execution Prompt as shown in Figure 3-9 on page 52. You are then prompted to specify a data set name and a name for the WSL to store the statements, as shown in Figure 3-64.

In this example, we used the **GEN** command as described in 5.4.1, “Using generate (GEN) to clone a database” on page 213, to generate the DDL from one of our databases, **DSG24D03** as input. We want to store the output from this generation into a WSL. We use the data set **ADMIN.WSL** and the WSL name **GEN00001**.

DB2 Admin creates a new data set and a new WSL if they do not already exist. If a WSL with that name already exists in the data set, the message “Existing Worklist” appears in the left corner. Specify whether to **APPEND** or to **REPLACE** the current contents.

```
DB2 Admin -------------- DB8A Generate SQL from DB2 catalog ------------- 17:38
Option ===>

 Generate SQL statements for database DSG24D03          DB2 System: DB8A
                                                        DB2 SQL ID: PAOLOR3

 SQL statement types to be generated from the DB2 catalog:
 EssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
 e DB2 Admin ------------------ Specify Work Statement List ----------------- e
 e Existing Worklist                                                          e
 e                                                                            e
 e  Work stmt list dsn  . . . 'ADMIN.WSL'                                     e
 e  Work stmt list name . . . GEN00001                                        e
 e                                                                            e
 e  Existing name action  . . .  REPLACE  (Append or Replace)                 e
 e                                                                            e
 e                                                                            e
 e                                                                            e
 e                                                                            e
 e                                                                            e
 e                                                                            e
 Dsssssssssssssssssssssssssssssssssssssss
```

```
DB2 Admin --------------- DB8A Manage Work Statement Lists -------------- 18:01
Option ===> 1

   1 - Show work statement list library                  DB2 System: DB8A
   2 - Show work statement list                          DB2 SQL ID: PAOLOR3

 Work stmt list dsn  ===> 'ADMIN.WSL'
 Work stmt list name ===> GEN00001
```

Figure 3-65 Manage Work Statement Lists

Note: You can directly jump to this panel from any screen in DB2 Admin and DB2 Compare by typing **WSL** as a primary command.

Selecting option **1** shows you the list of WSLs within the WSL data set. In the example shown in Figure 3-66, you see three different WSLs.

```
DB2 Admin ----------- Work Statement List Library: 'ADMIN.WSL' -- Row 1 from 3
Command ===>                                                Scroll ===> CSR

Line commands:
 S - Show  R - Run in batch  D - Delete  C - Copy  A - Append  Q - Clone
 I - Interpret  V - Validate  E - Edit

Sel Name      Size Created    Changed          ID
    *           * *           *                *
--- -------- ----- ---------- ---------------- --------
    GEN00001   847 2007/03/01 2007/03/01 18:22 PAOLOR3
    RSTATS01    17 2007/03/01 2007/03/01 18:24 PAOLOR3
    START001     3 2007/03/01 2007/03/01 18:21 PAOLOR3
****************************** END OF DB2 DATA ******************************
```

Figure 3-66 List WSL(s) in WSL Library

Selecting option **2**, specifying a WSL name in Figure 3-65 shows you the contents of that WSL, as shown in Figure 3-67. If the entered WSL data set or the WSL itself does not exist, DB2 Admin creates it for you.

There are a few operations that can be invoked to manipulate WSLs:

S Show the work statement list (see Figure 3-66 for an example).

R Generate a job to run a WSL in batch.

The statements from the WSL are used to create SQL statements, DB2 utility statements, and DB2 and DSN commands. An ADBTEP2 job can then run these commands and statements as a batch job.

D Delete this work statement list from the library.

C Copy this work statement list and append it to the work statement list, where you specify option “A”.

A Append the work statement list, selected by option “C” into this member (WSL).

Q Clone a WSL.

Cloning a WSL allows you to use an existing WSL to create a new WSL as a clone of that original WSL. Cloning is discussed in more detail in 3.9.7, “Cloning a work statement list” on page 105.

I Interpret the work statement list.

Interpreting a WSL allows you to selectively list the different SQL statements, DB2 commands, and utility statements that the WSL contains.

V Validate the work statement list.

Validating a WSL allows you to have the syntax of the SQL statements checked and to assess the impact that running the WSL would have on objects.

E Edit the work statement list.

```
DB2 Admin ----------- Show Work Statement List: GEN00001 -- Row 1 to 33 of 801
Command ===>                                                  Scroll ===> CSR

Line commands:
 D - Delete  I - Insert  E - Edit  C - Copy  M - Move  A - After  B - Before
 R - Repeat

Select Type Statement
       *    *
------ ---- ------------------------------------------------------------------>
       COM  -- EDITED BY PAOLOR3  ON 2007/03/01 AT 18:22
       COM  -- GENERATED BY PAOLOR3  ON 2007/03/01 AT 17:49
       COM  ------------------------------------------------------------------
       COM  --
       COM  -- Database 2 Administration Tool (DB2 Admin) , program 5697-L90
       COM  --
       COM  -- ADB2GEN  -  Extract object definitions from the DB2 Catalog tabl
       COM  --
       COM  -- Input prepared on : DB8A (810)     Extract time : 2007-03-01 17:
       COM  --
       COM  -- Catalog values overridden : none
       COM  --
       COM  -- Generate : SG=Y DB=Y TS=Y TB=Y VW=Y IX=Y SY=Y AL=Y LB=Y CM=Y FK=
       COM  --            TG=Y UT=N UF=N SP=N SQ=N
       COM  -- Grants   : SG=Y DB=Y TS=Y TB=Y VW=Y SC=N UT=N UF=N SP=N SQ=N
       COM  --
       COM  -- ADB2GEN: Generate DDL for Database DSG24D03
       COM  --
       COM  ------------------------------------------------------------------
       COM  -- Stogroup=GLWG01
       COM  ------------------------------------------------------------------
       COM  --
       DML    SET CURRENT SQLID='PAOLOR4'
       COM  --
       DDL    CREATE STOGROUP GLWG01..    VOLUMES(SBOX52, ..            SBOX58,
       COM  --
       DML    COMMIT
```

Figure 3-67 Show Work Statement List GEN00001 panel

Figure 3-67 shows part of the contents of WSL *GEN00001* after selecting option **S.** This WSL was generated as output from a GEN command (to reverse engineer a database).

3.9.4 Modifying a work statement list

Figure 3-67 shows three columns:

- SELECT
- TYPE
- STATEMENT

In the **SELECT** column you can enter the following commands to modify the WSL:

D Delete the statement from the list.

I Insert a statement into the list.

E Edit the statement.

C Copy this statement to a destination identified by a “A” (after) or a “B” (before) line command.

M Move this statement to a destination identified by a “A” (after) or a “B” (before) line command.

A To identify the destination of a move/copy operation is after this line.

B To identify the destination of a move/copy operation is before this line.

R Repeat the statement.

The **TYPE** column shows the type of statement. The following types are currently known:

COM Comment
DDL Data definition SQL statement
DCL Authorization related SQL statement
DML Data manipulation SQL statement
DB2 DB2 command
DSN DSN command
UTL Utility statement
ADM DB2 Admin statement

You can display only one type of statement at a time. In Figure 3-68 we display only DDL statements.

Archived

```
DB2 Admin ----------- Show Work Statement List: GEN00001 ----- Row 31 from 801
Command ===>                                                  Scroll ===> CSR

Line commands:
 D - Delete  I - Insert  E - Edit  C - Copy  M - Move  A - After  B - Before
 R - Repeat

Select Type Statement
       DDL  *
------ ---- ------------------------------------------------------------------>
       DDL    CREATE STOGROUP GLWG01..    VOLUMES(SBOX52, ..            SBOX58,
       DDL    CREATE DATABASE DSG24D03..    BUFFERPOOL BP15..     INDEXBP    BP1
       DDL    CREATE TABLESPACE GLWS001..    IN DSG24D03..    USING STOGROUP GL
       DDL    CREATE TABLE DSG24D03.GLWTACT..     (ACT_NO              INTEGER
       DDL    COMMENT ON TABLE DSG24D03.GLWTACT..    IS 'ACTivities that form p
       DDL    CREATE UNIQUE INDEX DSG24D03.GLWXACT1..    ON DSG24D03.GLWTACT..
       DDL    CREATE TABLE DSG24D03.GLWTJBS..     (JN_NO               SMALLIN
       DDL    COMMENT ON TABLE DSG24D03.GLWTJBS IS 'Employee JoBs and Salaries.
       DDL    CREATE UNIQUE INDEX DSG24D03.GLWXJBS1..    ON DSG24D03.GLWTJBS..
       DDL    CREATE TABLE DSG24D03.GLWTLCN..     (LOC_NO              SMALLIN
       DDL    COMMENT ON TABLE DSG24D03.GLWTLCN IS 'Department LoCatioNs.'
       DDL    CREATE UNIQUE INDEX DSG24D03.GLWXLCN1..    ON DSG24D03.GLWTLCN..
       DDL    CREATE TABLE DSG24D03.GLWTLNM..     (LN_NO               SMALLIN
       DDL    COMMENT ON TABLE DSG24D03.GLWTLNM IS 'Employee Last NaMes.'
       DDL    CREATE UNIQUE INDEX DSG24D03.GLWXLNM1..    ON DSG24D03.GLWTLNM..
```

Figure 3-68 Show only DDL Statements for WSL GEN00001

The third column **STATEMENT** contains the work statement itself.

To edit a statement, specify **E** next to the work statement. In the panel shown in Figure 3-69, you see the DDL after entering the option **E** next to a DDL work statement.

```
DB2 Admin ---------------- Edit Statement ---------------- Columns 00001 00072
Command ===>                                                  Scroll ===> CSR

****** ***************************** Top of Data ******************************
000001   CREATE TABLESPACE GLWS001
000002     IN DSG24D03
000003     USING STOGROUP GLWG01
000004     PRIQTY 16 SECQTY 48
000005     ERASE  NO
000006     FREEPAGE 0 PCTFREE 0
000007     GBPCACHE CHANGED
000008     TRACKMOD YES
000009     SEGSIZE 4
000010     BUFFERPOOL BP15
000011     LOCKSIZE ANY
000012     LOCKMAX SYSTEM
000013     CLOSE YES
000014     COMPRESS NO
000015     CCSID      EBCDIC
000016     DEFINE YES
000017     MAXROWS 255
****** **************************** Bottom of Data ****************************
```

Figure 3-69 Edit a DDL Work Statement

To add more work statements to an existing list, you can use the **I** line command on any line in Figure 3-67 where you want to add a work statement. This takes you to the panel in Figure 3-70.

```
DB2 Admin ------------------ DB8A Insert Work Statement ----------------- 19:18
Option ===>

Enter the type of statement you want to insert:

 1 - A comment statement
 2 - A definition SQL statement
 3 - A authorization SQL statement
 4 - A update SQL statement
 5 - A DB2 command
 6 - A DSN command
 7 - A DB2 utility statement
 8 - A DB2 Admin statement
```

Figure 3-70 Insert Work Statement

You can choose one of the eight options to create a new work statement. This takes you to an ISPF edit session, where you can insert an appropriate statement or command, like the one shown in Figure 3-71.

```
DB2 Admin ---------------- Edit Statement ---------------- Columns 00001 00072
Command ===>                                                 Scroll ===> CSR

****** ***************************** Top of Data ******************************
000100 REPAIR SET TABLESPACE DSG24D03.GLWSDPT NOCOPYPEND
****** **************************** Bottom of Data ****************************
```

Figure 3-71 Edit Statement

In this example we added a statement for the REPAIR utility. After pressing the Enter key, DB2 Admin adds this work statement to the WSL. DB2 Admin also adds the right statement type.

Note: DB2 Admin maintains ISPF statistics when a WSL is created or updated. DB2 Admin supports ISPF statistics for members of a Work Statement List (WSL) for frequently used DB2 Admin functions.

Figure 3-72 shows WSL *GEN00003*, where all possible work statements have been added with examples.

```
DB2 Admin --------- Show Work Statement List: GEN00003 ----- Row 1 to 14 of 14
Command ===>                                                  Scroll ===> CSR

Valid line commands are:
 D - Delete  I - Insert  E - Edit  C - Copy  M - Move  A - After  B - Before
 R - Repeat

Select Type Statement
       *    *
------ ---- ------------------------------------------------------------------
*      COM            OPTION 2: DDL
*      DDL  CREATE INDEX DSG24D03.GLWXEMP1 ..
*      COM            OPTION 3: DCL
*      DCL  GRANT SELECT ON TABLE DSG24D03.GLWTEMP TO PUBLIC
*      COM            OPTION 4: UPDATE SQL
*      DML  UPDATE DSG24D03.GLWTEMP SET LASTNAME = 'NEW' WHERE EMPNO='000010'
*      COM            OPTION 5: DB2 COMMAND
*      DB2  -DIS THD(*)
*      COM            OPTION 6: DSN COMMAND
*      DSN  REBIND PLAN(ADBBETA2)
*      COM            OPTION 7: UNLOAD STATEMENT
*      ADM  PARALLEL UNLOAD
*      ADM  JOB
*      DB2  -STA DB(DSG24D03) SPACE(GLWSEMP) ACCESS(RO)
*      DML  TSODELETE 'PAOLOR3.OC.DEMO.CNTL.TBL0001'
*      DML  TSODELETE 'PAOLOR3.OC.DEMO.UNLD.TBL0001'
*      UTL  TEMPLATE UTLPUNCH DSN 'PAOLOR3.OC.DEMO.CNTL.TBL0001'..         UNIT
*      UTL  TEMPLATE SYSREC DSN 'PAOLOR3.OC.DEMO.UNLD.TBL0001'..         UNIT S
*      UTL  UNLOAD DATA FROM TABLE "DSG34D03"."GLWTDPT" PUNCHDDN(UTLPUNCH)
*      DML  TSODELETE 'PAOLOR3.OC.DEMO.CNTLC.TBL0001';..TSODELETE 'PAOLOR3.OC.D
*      TSO  ALLOC DD(DDLIN) DUMMY
*      TSO  ALLOC DD(CNTLI001)..      DS('PAOLOR3.OC.DEMO.CNTL.TBL0001') SHR
*      TSO  ALLOC DD(CNTLO001)..      DS('PAOLOR3.OC.DEMO.CNTLC.TBL0001')..
*      TSO  ALLOC DD(DATAI001)..      DS('PAOLOR3.OC.DEMO.UNLD.TBL0001') SHR
*      TSO  ALLOC DD(DATAO001)..      DS('PAOLOR3.OC.DEMO.UNLDC.TBL0001') USING
*      ADM  ADMIN ALTER CONVERT DSG24D03.GLWTDPT.. NEWNAME DSG24D03.GLWTDPTX
*      ADM  ENDJOB
****************************** END OF DB2 DATA ******************************
```

Figure 3-72 Show Work Statement List GEN00003

Note: There is currently no *modeling* for the various types of work statements. You can create a *model* WSL for each statement type yourself and store them in a WSL. When required, you can copy them into your newly defined WSL.

3.9.5 Examining a work statement list

Before you execute a WSL, you might want to examine the contents of the WSL and assess the impact that running the WSL would have on objects. DB2 Admin provides either an *interpret* or *validate* option.

Interpret

Interpreting a WSL allows you to selectively list the different SQL statements, DB2 commands, and utility statements that the WSL contains.

Issue the **I** command on the Work Statement List Library panel (see Figure 3-66 on page 94). We want to interpret the WSL GEN00001. As you can see on the Interpret Work Statement List Options panel in Figure 3-73, we choose each statement type.

```
DB2 Admin ------------ Interpret Work Statement List Options ------------------

 Specify S to select Work Statement List statement types:

   SQL:                DB2 Utilities:              DB2 Commands:
     S DDL               S Load/Unload               S Plan/packages
         ALTER               LOAD                        BIND
         CREATE              UNLOAD/REORG UNLOAD         REBIND
         DROP            S Backup/Recovery               FREE
         COMMENT ON          COPY                    S Other
         LABEL ON            COPYTOCOPY                  RUN
         SET                 MERGECOPY                   START/STOP
     S DCL                   MODIFY                      Other
         GRANT               QUIESCE               Admin:
         REVOKE              REBUILD                 S Data set
     S DML                   RECOVER                     ALLOC
         DELETE              REPORT                      TSODELETE
         INSERT          S Other                         LISTDEF
         UPDATE              CHECK                       TEMPLATE
       Other                 DIAGNOSE                    ADBSYSIN
         COMMIT              REORG                   S Other
         Comments            REPAIR                      ADBPAUSE
       S Other               RUNSTATS                    UTILFROM
                             STOSPACE                    REXX Execs
                             Other                       Other
```

Figure 3-73 Interpret WSL Options panel

Archived

When you press Enter, the Interpret Work Statement List report is generated, as shown in Figure 3-74. Use the **S** line command to show an object, or **V** to view or format a statement.

```
DB2 Admin --------- Interpret Work Statement List: GEN00001 Row 1 to 34 of 119
Command ===>                                                  Scroll ===> CSR

Line commands:  S - Show object  V - View statement

Sel   Seq Action     Object Type   Qual     Name                Note
    *     *          *             *        *                   *
--- ----- ---------- ------------- -------- ------------------ -------------
       30 SET        SQLID         PAOLOR4
       32 CREATE     STOGROUP               GLWG01
       41 SET        SQLID         PAOLOR2
       43 CREATE     DATABASE               DSG24D0C
       45 SET        SQLID         PAOLOR3
       47 GRANT      DATABASE               DSG24D0C
       56 SET        SQLID         PAOLOR2
       58 CREATE     TABLESPACE    DSG24D0C GLWS001
       67 SET        SQLID         DSG24D0C
       69 CREATE     TABLE         DSG24D03 GLWTACT
       71 COMMENT ON TABLE         DSG24D03 GLWTACT
       80 CREATE     INDEX         DSG24D03 GLWXACT1          Unique
       88 CREATE     TABLE         DSG24D03 GLWTJBS
       90 COMMENT ON TABLE         DSG24D03 GLWTJBS
       99 CREATE     INDEX         DSG24D03 GLWXJBS1          Unique
      107 CREATE     TABLE         DSG24D03 GLWTLCN
      109 COMMENT ON TABLE         DSG24D03 GLWTLCN
      118 CREATE     INDEX         DSG24D03 GLWXLCN1          Unique
      126 CREATE     TABLE         DSG24D03 GLWTLNM
      128 COMMENT ON TABLE         DSG24D03 GLWTLNM
```

Figure 3-74 Interpret WSL Report panel

Validate

Validating a WSL allows you to have the syntax of the SQL statements checked and to assess the impact that running the WSL would have on objects.

The impact analysis portion of the validate report lists the impact to the objects by these categories:

- **Implicitly dropped objects:**

  Existing objects that are implicitly dropped but not re-created by the WSL.
- **Explicitly dropped objects:**

  Existing objects that are explicitly dropped but not re-created by the WSL.
- **Recreated objects:**

  Existing objects that are implicitly or explicitly dropped and re-created by the WSL.
- **Altered objects:**

  Existing objects that are altered by the WSL.
- **Created objects:**

  Objects that did not exist and are created by the WSL.
- **Temporary objects:**

  Objects that did not exist and are created and then dropped by the WSL.

Archived

To validate a WSL, issue the **V** command on the Work Statement List Library panel (Figure 3-66 on page 94). We are using again the WSL GEN00001. The JCL to generate the batch job to produce the Validate Work Statement List report is displayed (see Figure 3-75).

```
DB2 Admin -------------- Edit Generated JCL -------------- Columns 00001 00072
Command ===>                                                 Scroll ===> CSR

****** ***************************** Top of Data ******************************
000001 //PAOLOR3# JOB (999,POK),'ITSO CM RESIDENCY',
000002 //         REGION=0M,MSGCLASS=X,CLASS=A,
000003 //         MSGLEVEL=(1,1),NOTIFY=&SYSUID
000004 /*JOBPARM S=SC63
000005 //*
000006 //*************************************************************ADB2SPFB**
000007 //* DB2 ADMIN ISPF BATCH
000008 //*********************************************************************
000009 //WSLVAL   EXEC PGM=IKJEFT01,
000010 // PARM='ISPSTART CMD(%ADB2WVL)',
000011 // REGION=0M,
000012 // DYNAMNBR=100
000013 //SYSEXEC  DD DISP=SHR,DSN=GOC.V7R2M0.SGOCEXEC
000014 //         DD DISP=SHR,DSN=ADB.V7R2M0.SADBEXEC
000015 //SYSPROC  DD DISP=SHR,DSN=GOC.V7R2M0.SGOCCLST
000016 //         DD DISP=SHR,DSN=ADB.V7R2M0.SADBCLST
000017 //ISPLLIB  DD DISP=SHR,DSN=GOC.V7R2M0.SGOCLLIB
000018 //         DD DISP=SHR,DSN=ADB.V7R2M0.SADBLLIB
000019 //         DD DISP=SHR,DSN=ISP.SISPLOAD       ISPF BASE
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 44 Line(s) not Displayed
000064 //*********************************************************************
000065 //* END OF ISPF BATCH STEP SET UP
000066 //*********************************************************************
000067 //*************************************************************ADB2WVAL *
000068 //ADBWL    DD DISP=SHR,
000069 //         DSN=ADMIN.WSL
000070 //CAT      DD DSN=&&CATOUT2,
000071 //         DCB=(LRECL=16800,RECFM=VB,DSORG=PS),
000072 //         SPACE=(TRK,(15,15),RLSE),
000073 //         UNIT=SYSDA,
000074 //         DISP=(NEW,PASS)
000075 //IN       DD *
000076   DB2SYS    = DB8A,
000077   DB2ALOC   =
000078   DB2SERV   = DB8A
000079   DB2AUTH   = PAOLOR3
000080   DB2REL    = 815
000081   ADBTEST   = YES,
000082   PLAN      = A72,
000083   SRCWSLST  = GEN00001,
000084   SRCWSLIB  = ADMIN.WSL
000085 /*
****** **************************** Bottom of Data ****************************
```

Figure 3-75 Generated JCL for WSL validation

After submitting the JCL, the Validate Work Statement List report is generated and displayed, as shown in Figure 3-76.

```
-------------------------------------------------------------------------------
ADB2WVL - Validate Work Statement List
-------------------------------------------------------------------------------

          DB2 Administration Tool
          5697-K90 (C) Copyright IBM Corporation 2001, 2004.
          All rights reserved.  Licensed materials - property of IBM.
          US Government Users Restricted Rights - Use, duplication or disclosure
          restricted by GSA ADP schedule contract with IBM Corp.

-------------------------------------------------------------------------------
REFERENCE FOR CATALOG OBJECT STATUS
-------------------------------------------------------------------------------
IMPLICITLY DROPPED OBJECTS - Existing catalog objects that are implicitly
                             dropped and not recreated by the WSL.
TEMPORARY OBJECTS          - Objects that are created and dropped during
                             execution of the WSL.  Temporary objects do not
                             exist in the catalog before or after WSL execution.
CREATED OBJECTS            - Objects that are created by the WSL that did not
                             exist in the catalog.
EXPLICITLY DROPPED OBJECTS - Existing catalog objects that are explicitly
                             dropped and not recreated by the WSL.
ALTERED OBJECTS            - Existing catalog objects that are modified by
                             ALTER statements in the WSL.
RECREATED OBJECTS          - Existing catalog objects that are implicitly or
                             explicilty dropped and later recreated by the WSL.
-------------------------------------------------------------------------------


VALIDATE WORK STATEMENT LIST REPORT
===================================

Prepared on DB8A (DB2 Release 815) by PAOLOR3 at 2007-03-05 14:04
for ADMIN.WSL(GEN00001)

Error processing Stogroup  GLWG01 in a CREATE statement:Object already exists
Error processing Table DSG24D03.GLWTACT in a CREATE statement:Object already exists
Error processing Index DSG24D03.GLWXACT1 in a CREATE statement:Object already exists
Error processing Table DSG24D03.GLWTJBS in a CREATE statement:Object already exists
Error processing Index DSG24D03.GLWXJBS1 in a CREATE statement:Object already exists
Error processing Table DSG24D03.GLWTLCN in a CREATE statement:Object already exists
Error processing Index DSG24D03.GLWXLCN1 in a CREATE statement:Object already exists
Error processing Table DSG24D03.GLWTLNM in a CREATE statement:Object already exists
Error processing Index DSG24D03.GLWXLNM1 in a CREATE statement:Object already exists
.
.
.
CREATED OBJECTS
 ----------------
   Table Space DSG24D0C.GLWS001
   Database  DSG24D0C
```

Figure 3-76 Validate Work Statement List report

3.9.6 Executing a work statement list

To create a job that executes this WSL, use the **R** line command in front of a WSL. In our example, this is job RSTATS01, as shown in Figure 3-77.

```
DB2 Admin ----------- Work Statement List Library: 'ADMIN.WSL' -- Row 1 from 3
Command ===>                                                  Scroll ===> CSR

Line commands:
 S - Show  R - Run in batch  D - Delete  C - Copy  A - Append  Q - Clone
 I - Interpret  V - Validate  E - Edit

Sel Name      Size Created    Changed          ID
    *            * *          *                *
--- -------- ----- ---------- ---------------- --------
    GEN00001   848 2007/03/01 2007/03/01 19:35 PAOLOR3
R   RSTATS01    17 2007/03/01 2007/03/01 18:24 PAOLOR3
    START001     3 2007/03/01 2007/03/01 18:21 PAOLOR3
****************************** END OF DB2 DATA ******************************
```

Figure 3-77 Generate job for Work Statement List RSTATS01

The job shown in Example 3-2 is generated.

Example 3-2 Job to execute WSL TEST01A

```
DB2 Admin -------------- Edit Generated JCL -------------- Columns 00001 00072
 Command ===>                                                 Scroll ===> CSR

 ****** ****************************** Top of Data ******************************
 000001 //PAOLOR3# JOB (999,POK),'ITSO CM RESIDENCY',
 000002 //*       RESTART=STEPNAME, <== FOR RESTART REMOVE * AND ENTER STEP NAME
 000003 //         REGION=0M,MSGCLASS=X,CLASS=A,
 000004 //         MSGLEVEL=(1,1),NOTIFY=&SYSUID
 000005 /*JOBPARM S=SC63
 000006 //*******************************************************************
 000007 //*
 000008 //* DB2 BATCH MONITOR
 000009 //*
 000010 //* DB2 ADMIN GENERATED BATCH JOB.
 000011 //*
 000012 //*************************************************************ADB2WL4**
 000013 //DB2B EXEC PGM=IKJEFT01,DYNAMNBR=100
 000014 //STEPLIB  DD DISP=SHR,DSN=DB8A8.SDSNEXIT
 000015 //         DD DISP=SHR,DSN=DB8A8.SDSNLOAD
 000016 //SYSEXEC  DD DISP=SHR,DSN=GOC.V7R2M0.SGOCEXEC
 000017 //         DD DISP=SHR,DSN=ADB.V7R2M0.SADBEXEC
 000018 //MSGLIB   DD DISP=SHR,DSN=GOC.V7R2M0.SGOCLLIB
 000019 //         DD DISP=SHR,DSN=ADB.V7R2M0.SADBLLIB
 000020 //SYSTSPRT DD SYSOUT=*
 000021 //SYSPRINT DD SYSOUT=*
 000022 //UTPRINT  DD SYSOUT=*
 000023 //RNPRIN01 DD SYSOUT=*
 000024 //ADBDIAG  DD SYSOUT=*
 000025 //SYSTSIN  DD *
 000026  DSN SYSTEM(DB8A)
 000027  RUN  PROGRAM(ADBTEP2) PLAN(A72) -
 000028    LIB('ADB.V7R2M0.SADBLLIB')  -
 000029    PARMS('/WORKLIST(RSTATS01) SSID(DB8A)  -
```

```
000030  RESTART(YES),MAXE(10)')
000031  END
000032 //SYSIN    DD *
000033 -- Created by PAOLOR3 on 2007/03/01 at 18:24 by cloning of
000034 -- source work stmt LIST  STATS001  FROM library  'ADMIN.WSL'
000035 RUNSTATS TABLESPACE "DSG24D03"."GLWSDPT"
000036   INDEX
000037    (
000038     ALL
000039    )
000040   SHRLEVEL CHANGE;
000041 /*
```

3.9.7 Cloning a work statement list

Using option **Q** in the Work Statement List Library panel, as shown in Figure 3-66 on page 94, you have the possibility to clone a WSL. Cloning a WSL allows you to use an existing WSL to create a new WSL.

In the Clone Work Statement List panel, shown in Figure 3-78, a new WSL name must be specified.

```
DB2 Admin ------- Clone Work Statement List -----------------------------------
Command ===>                                                  Scroll ===> CSR

 Input work stmt list information:                      DB2 System: DB8A
   Work stmt list . . . : GEN00001                      DB2 SQL ID: PAOLOR3
   from library . . . . : 'ADMIN.WSL'

 Output work stmt list information:
   Library (PDS name) . :
   Work stmt list . . . : GEN0001C (will be new PDS member)

 Execution mode . . . . : TSO      (BATCH or TSO)
   PDS for batch jobs . :
   PDS member . . . . . :
   Unit type  . . . . . : SYSDA

 Use Masking. . . . . . : YES      (Yes/No)
   Apply masking to data set names. . :        (Yes/No=default)

 Use local DB2 catalog information to replace: (Yes/No)
   Authorizations . . . . . . . . . . : NO
   Partitioning attributes  . . . . . : NO
   Table space and index attributes . : NO

 Optional overwrites for table space or index space attributes
   Overwrites data set. :
   Edit data set  . . . : (Yes/No=default)

 Additional parameters:
   Message output file  : 'ADMIN.OUTPUT'

 M - Browse message output file
```

Figure 3-78 Clone Work Statement List panel

For the new cloned WSL, we used the name **GEN0001C**.

> **Note:** The new WSL name cannot be the same as the existing WSL (the one you are cloning from), even if you clone a WSL into another library. Each WSL is an ISPF table with a unique name. Therefore, the WSL you are copying from must be named differently from the WSL you are copying to.

In DB2 Admin you can specify a mask when you generate SQL statements from the DB2 catalog, to reverse engineer DB2 objects, or migrate objects, data, or catalog statistics.

In our sample we use masking during the clone process to change object names, from the original WSL. The original names in the source WSL remain unchanged.

You can optionally *overwrite* the current values of the table space and index space attributes (for example, increase the primary quantity (PRIQTY) and secondary quantity (SECQTY) when they appear in SQL statements CREATE/ALTER TABLESPACE/INDEX of the new WSL). In this case you have to specify YES, the values in the original WSL are kept unchanged.

You can optionally *override* the specified values of the table space and index space primary quantity (PRIQTY) and secondary quantity (SECQTY) when they appear in CREATE/ALTER TABLESPACE/INDEX SQL statements of the WSL. These overrides can be done by opting to use values from the local DB2 catalog, or by specifying an overwrite data set.

You are able to override the authorizations to objects created by the WSL with grants for the local DB2 catalog. With the Partitioning override function you can override the characteristics of table spaces and indexes which are partitioned in the local DB2 catalog.

The values for these overrides are taken from the local DB2 catalog tables where the cloning job is executing. Therefore make sure, when using overrides, to clone the WSL on the target system (where the cloned WSL is going to be executed). Overrides are done after masks are applied.

For more information about masking, refer to 4.9, “Using the masking feature” on page 166.

As we have specified **YES** for using masks, the pop-up panel shown in Figure 3-79, appears.

```
DB2 Admin ----------------------- Specify Mask ----------------------------
Command ===>

Mask Table Entry:
  Owner . . . . PAOLOR3  >            (? to look up)
  Name  . . . . GEN0001M           >  (? to look up)
Data Set:
 Mask DSN . . .
Options:
 Edit Mask  . . YES                (Yes/No)
```

Figure 3-79 Specify Mask Table or Data Set

You can define a mask either in a data set, or in a the CM database. As we have Change Management in DB2 Admin enabled, we add a table entry in the CM database.

Tip: From the control aspect, consider managing all your masks through Change Management. Masks that are specified when you import changes through CM must be defined in the CM database. Masks are stored in the CM database tables ADBCMASK, and ADBCMASKS.

The CM Insert Mask panel, as shown in Figure 3-80, allows you to insert a mask into the CM tables.

```
DB2 Admin ---------------- CM - Insert Mask ---------------------------- 17:14
Command ===>


Owner . . . PAOLOR3 >  (Optional, default is PAOLOR3)
Name  . . . GEN0001M >  (Required, ? to look up)
Comment . . Mask for cloning WSL GEN00001     >
```

Figure 3-80 CM Insert Mask panel

After pressing Enter, the insert statement is executed. Press PF3 to get to the CM Mask Lines panel, as shown in Figure 3-81.

```
DB2 Admin ----------------- CM - Mask Lines -------------------- Row 1 from 1
Command ===>                                              Scroll ===> CSR

Mask lines for mask "PAOLOR3"."GEN0001M"
Commands: SAVE
Line commands:
  I - Insert  D - Delete  R - Repeat  M - Move  A - After  B - Before

Sel    Sequence Type     From                 To                     Oper.  T
             * *        *                    *                      *      *
--- ----------- -------- --------------------> ----------------------> ------ -

*            1 TBNAME   GLW*                 CLO*                   UPDATE
****************************** END OF DB2 DATA ******************************
```

Figure 3-81 CM Mask Lines panel

Archived

The Mask Lines panel displays information about the mask lines for the mask that you selected. You can specify one ore more masks. Notice that mask names have a hierarchy.

There are several panel columns on the CM Mask Lines panel shown in Figure 3-77 on page 104.

Select:
Input field where you can enter one of the listed line commands.

Sequence:
The order in which the masking occurs.

Type:
The valid mask fields, including hierarchy, are:

- COLNAME
- NAME
- DBNAME,TSNAME,TBNAME,IXNAME,UDFNAME,
- UDTNAME,COLLNAME,PKGNAME,PGMNAME,PLNNAME,
- DBRMNAME,STPNAME,SFNAME,TGNAME,GRPNAME,
- VCATNAME,GBPNAM
- BPNAME
- TSBPNAME,IXBPNAME
- SGNAME
- TSSGNAME,IXSGNAME
- AUTHID
- OWNER,SQLID,SCHEMA
- GRANTID
- GRANTOR,GRANTEE

From:
The string or pattern to match for the object.

To:
The replacement value for any found occurrences of 'from'.

Oper.:
The last operation that was performed on the row: insert, update, or move.

T:
Indicates a pending move operation:

M - A pending move exists
blank - No pending move exists

We want to define a mask that translates any table name that starts with **GLW** to a name that starts with **CLO**. After issuing the **SAVE** primary command, the insert statement is executed. Press PF3 to return to Clone Work Statement List panel.

The following message appears on the top of the panel:

```
Edit of mask done. Press <enter> now to proceed with cloning or next function.
```

After pressing Enter, the cloning starts and creates a new member in the output WSL library, specified on the panel in Figure 3-78 on page 105.

When the cloning operation ends successfully, you receive a message like the one shown in Figure 3-82.

```
DB2 Admin ------- Clone Work Statement List -----------------------------------
Command ===>                                                  Scroll ===> CSR
GEN00001  was successfully cloned AS  GEN0001C in  'ADMIN.WSL' .
 Input work stmt list information:                       DB2 System: DB8A
   Work stmt list . . . : GEN00001                       DB2 SQL ID: PAOLOR3
   from library . . . . : 'ADMIN.WSL'

 Output work stmt list information:
   Library (PDS name) . : 'ADMIN.WSL'
   Work stmt list . . . : GEN0001C (will be new PDS member)

 Execution mode . . . . : TSO      (BATCH or TSO)
   PDS for batch jobs . :
   PDS member . . . . . :
   Unit type  . . . . . : SYSDA

 Use Masking. . . . . . : YES      (Yes/No)
   Apply masking to data set names. . : NO     (Yes/No=default)

 Use local DB2 catalog information to replace: (Yes/No)
   Authorizations . . . . . . . . . . : NO
   Partitioning attributes  . . . . . : NO
   Table space and index attributes . : NO

 Optional overwrites for table space or index space attributes
   Overwrites data set. :
   Edit data set  . . . :                     (Yes/No=default)

 Additional parameters:
   Message output file  : 'ADMIN.OUTPUT'

 M - Browse message output file
```

Figure 3-82 Successful generation of WSL GEN0001C

To view the cloned WSL, put an **S** next to the new WSL *GEN0001C* on the Work Statement List Library panel (see Figure 3-66 on page 94).

Archived

Figure 3-83 shows the contents of the newly created clone “GEN0001C”.

```
DB2 Admin ----------- Show Work Statement List: GEN0001C --- Row 6 to 37 of 765
Command ===>                                                  Scroll ===> CSR

Line commands:
 D - Delete  I - Insert  E - Edit  C - Copy  M - Move  A - After  B - Before
 R - Repeat

Select Type Statement
       *    *
------ ---- ------------------------------------------------------------------>
       COM  ------------------------------------------------------------------
       COM  --
       COM  -- Database 2 Administration Tool (DB2 Admin) , program 5697-L90
       COM  --
       COM  -- ADB2GEN  -  Extract object definitions from the DB2 Catalog tabl
       COM  --
       COM  -- Input prepared on : DB8A (810)     Extract time : 2007-03-01 17:
       COM  --
       COM  -- Catalog values overridden : none
       COM  --
       COM  -- Generate : SG=Y DB=Y TS=Y TB=Y VW=Y IX=Y SY=Y AL=Y LB=Y CM=Y FK=
       COM  --            TG=Y UT=N UF=N SP=N SQ=N
       COM  -- Grants   : SG=Y DB=Y TS=Y TB=Y VW=Y SC=N UT=N UF=N SP=N SQ=N
       COM  --
       COM  ------------------------------------------------------------------
       COM  --
       COM  -- ADB2GEN: Generate DDL for Database DSG24D0C
       COM  --
       COM  ------------------------------------------------------------------
       COM  ------------------------------------------------------------------
       COM  --    Table=DSG24D0C.GLWTACT In DSG24D03.GLWS001
       COM  ------------------------------------------------------------------
       COM  --
       COM  --
       DDL    CREATE TABLE DSG24D03.CLOTACT..     (ACT_NO INTEGER NOT NULL GENE
       COM  --
       DDL    COMMENT ON TABLE DSG24D03.CLOTACT ..    IS 'ACTivities that form
       COM  --
       DML    COMMIT
       COM  --
       COM  ------------------------------------------------------------------
```

Figure 3-83 Show Work Statement List GEN0001C

Error message processing

If an error occurs during cloning operation, a message is written to the data set named *<prefix>.CLONE.SYSPRINT.name-of-source-WSL* that DB2 Admin automatically creates.

You can browse the clone message data set for a WSL at anytime from the Clone Work Statement List panel, by entering `M` on the command line.

Error message processing in BATCH mode is reported in the SYSPRINT data set.

4

Working with the Change Management functions

In this chapter we provide information on the new Change Management (CM) functions that are available with V720 of DB2 Admin and DB2 Object Compare. This is a substantial enhancement to the products. We go through the concept of Change Management, new wording, cover the CM processes, and get an example of how everything comes together to accomplish a designated change task.

We recommend that you first get used to the various processes and terminology of Change Management before you start to implement your Change Management strategy in your DB2 environment. Every process ends up in a special status of your change; we describe these as well. We also inform you about navigating with your changes and how to use the features, *masking* and *ignore*.

We cover the following topics:

- General considerations
- Change navigation
- Change processes
- Change status
- Change types
- Cancel a change
- Restarting
- Using the masking feature
- Using the ignore feature
- Change terminology
- Versions

© Copyright IBM Corp. 2007. All rights reserved.

4.1 General considerations

Before you start learning about Change Management, we want to make you aware of some general topics that might help you in understanding Change Management.

4.1.1 Point-and-shoot

Many panels within Change Management are defined with point-and-shoot capability. You can position your cursor at those fields (by using the Tab key) and press Enter; this has the same effect as typing the command directly as a line command. For instance, the line commands CONTINUE and SAVE are probably the most frequently used candidates for point-and-shoot. Use the Tab key to position the cursor under the word (such as CONTINUE) and press Enter. This is the same as keying in the word `CONTINUE` on the command line.

4.1.2 Data sets

Many data sets are created during your use of the products Admin and Compare, which allocate libraries always with SMS defaults. Check your libraries being allocated to make sure that they do not belong to default management classes. If so, add appropriate ACS routines to your environment.

4.1.3 Version files

An important part of the product is based on versions; they are discussed in detail in this book. You might experience some confusion in terms of version terminology in the official literature. In previous days of DB2 Admin and DB2 Compare, versions already existed. They were saved in a data set and the terminology *version file* became a standard. These files can grow large in size, and some customers have found disk space relief by allocating those files to a data class that invokes SMS compression functions. They have also reduced elapsed time by explicitly specifying more I/O buffers.

With the introduction of Change Management, versions can now be stored within the Change Management database. But you might find that those versions stored internally in a DB2 table of the Change Management database are called *version files* anyway, although there is no connection to a data set.

Versions can be created in one of three ways:

- When using CM, you can define a version scope (which includes the objects to be included in a version) and then use the **`GV`** line command on the Version Scope panel to generate a version based on the scope.
- When running a change in CM, you can request a version of the objects be generated after the changes have been applied.
- When using DB2 Object Comparison Tool, you can have versions of the source and target objects generated (these are the versions saved as a data set). When CM is enabled, you have the option of storing versions in data sets or in the CM database.

4.2 Overview of Change Management

The Change Management function in DB2 Admin 7.2 enables the recording and tracking of changes and greatly enhances the promotion of changes that you make to your DB2 objects. These changes can be very complex, especially when others have defined changes that have yet to be run.

Before we briefly overview some of the new functionality, we want to set the boundaries about what a DB2 Change Management tool can do and what it cannot do. In fact, all tools in that area fall into this category (meaning that it applies also to non-IBM solutions).

First of all, DB2 Change Management is a concept that requires discipline to follow its rules. Buying and installing a tool is not enough to get Change Management done.

Change Management requires establishing rigid procedures about how a change is defined, executed, and propagated. No tool can enforce Change Management, users must follow an agreed procedure and the flow of actions. Tools accompany the flow of actions, and it is the users' task to define the Change Management steps and carve out a Change Management procedure. Before embarking on using Change Management (CM), the users should have a clear understanding of the desired change processes and have a change strategy defined.

One of the most important factors in determining a Change Management strategy is how a change request arrives, because this already represents the definition of the change:

- E-mail, phone, or written document
- DDL (the new structure) being assembled or generated
- AL, ALC, AL/RDEF, AL/RENAME within DB2 Admin
- COMPARE to identify differences as being the change request
- CREATE/DROP statements within DB2 Admin
- CM function change statements are inserted
- Work statement lists (currently not supported by CM)

Here are some other possible questions to be answered:

- What are the naming differences across environments?
- Which differences between the environments are wanted and which must be ignored?
- Are the objects in the various environments exactly matching?
- Shall every single change be propagated separately, or shall changes be batched together?
- Where is the master database that reflects the actual structure?

The change request then must be fed and mapped into the change procedure. One area that often gets overlooked is the handling of new objects. This is a change too within the database structure. New objects must be aligned at some stage to the standards within the customer environment (such as PRI QTY, SECQTY, COMPRESS YES/NO, and so on).

After answering such questions, one can start to design a Change Management process flow. Any attempt to start without a concept is only asking for problems later on in the deployment of changes.

To clarify this concept: Every customer implements change in a different way — and this is why there is not a fixed path that must be taken. The customer has to arrange the existing building blocks to accommodate their own requirements. However, every change finally feeds into the core functions later.

So what is a change anyway? It is the modification (or addition) of (or to) an existing structure in DB2, and can be of any level: database, table space, table, index, view, trigger, or column.

To the user, a change is basically a three-step process:

1. Define the change.
2. Generate the change.
3. Execute the change.

The user has to follow this sequence with any kind of tool. This fact dictates also how DB2 Admin and Compare support these events with their long established functions, available in previous releases, as shown in Figure 4-1:

1. ALTER (using AL, ALC, AL/RDEF, AL/RENAME), Compare
2. Generate a WSL or JCL
3. Execute the WSL or JCL

An occasional fourth step could be to restore an altered table to its original state prior to the change.

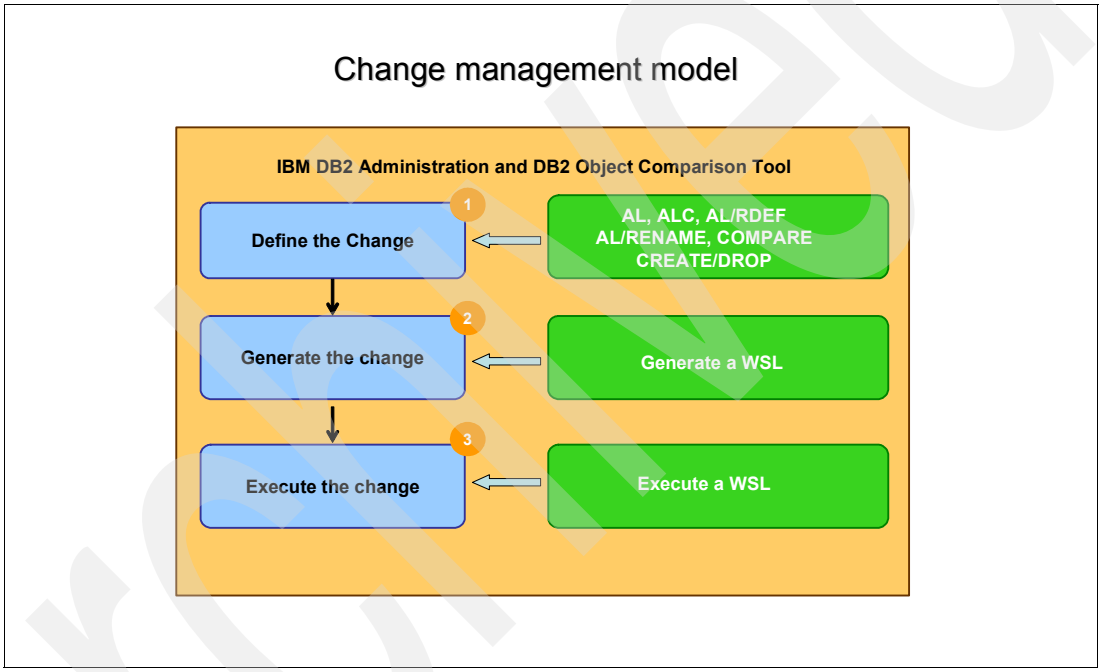

Figure 4-1 Change Management model before the new CM functions

However, in the past, all these functions were loosely tied together. There was no strict enforcement on these steps. Change Management now establishes formalized processes that reflect these actions (REGISTER, ANALYZE, RUN). One has to follow this process chain in order to get a change done. In fact, the REGISTER process is new and has been added to the core functionality.

Registration of a change is key, since formerly, there was no information available on the change itself. There was no control so that other users could see what changes are going on, which objects are affected. Now, the new Change Management supports the core functionality (REGISTER, ANALYZE, and RUN) as shown in Figure 4-2.

Everything centers around these core steps. This sounds simple, but actually can get very complicated if there are multiple versions of an object and multiple database instances.

For that reason, there are mechanisms in place that support such multiple environment requirements to propagate changes across multiple database instances. One of the major enhancements made was the introduction of the so-called delta change. This represents the net change and acts as a vehicle for transporting changes across multiple environments.

Additionally, many users might be working in parallel on different changes (under change IDs, which are unique inside the CM database). Every change (even from the same user) can affect another change, and this is also taken into consideration. And this is where PENDING CHANGES come into play.

PENDING CHANGES are registered changes in the CM database (with status DEFINED or ANALYZED) but not yet executed. Every subsequent change (with a different change ID) that comes after and that affects an object for which a change has been registered, gets asked what to do with this already PENDING CHANGE.

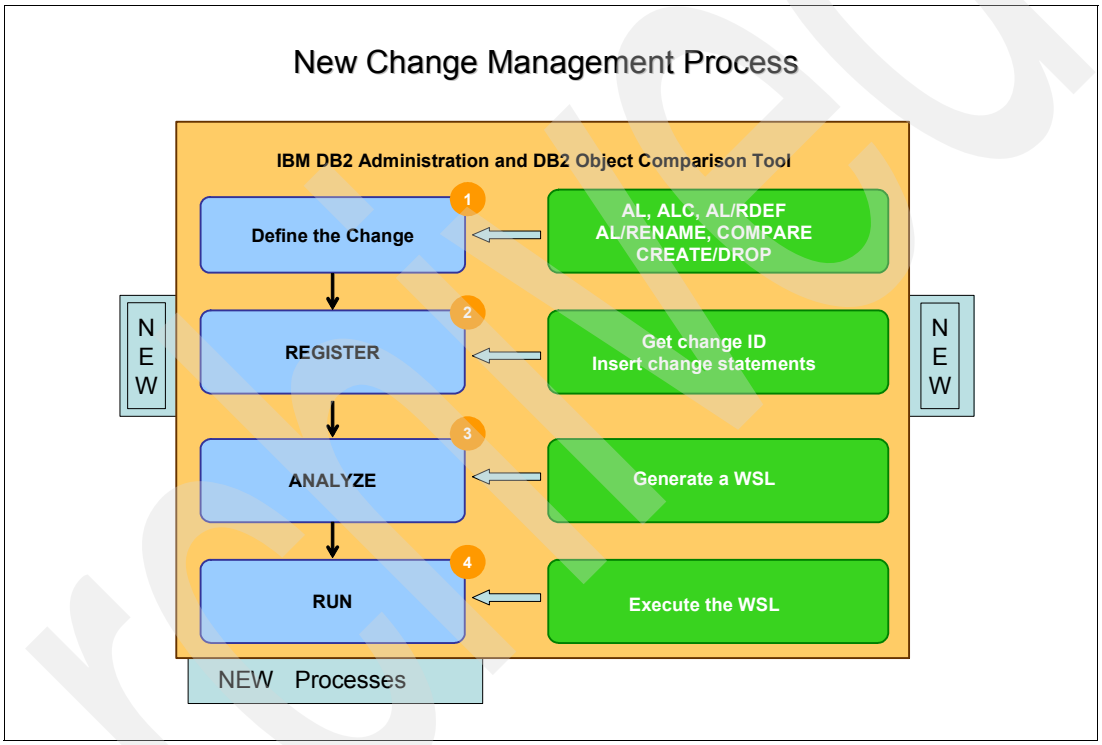

Figure 4-2 New Change Management Process

Example

USERA defines and registers a change on TABLE A. But USERA does not implement the change. Later, USERB also wants to change TABLE A. Before USERB can actually define the change, he is notified that there are PENDING CHANGES for TABLE A. CM also asks USERB how the new change should be handled (Applied or Superseded) with the existing change from USERA.

Another very important function that comes with this is the implementation of VIRTUAL CHANGES. Users pulling up a screen to define a change are prompted if their object affects an already defined change (meaning REGISTERED and also not yet done) . The idea here is to show a complete picture of an object as if these PENDING CHANGES had already been applied. This shows any change that is currently in progress by any user. — hence the name, virtual changes.

4.2.1 Change Management (CM) offers a structured way to work with changes

All this requires a controlled and manageable fashion because these functions were either independent from each other before or simply did not exist (such as PENDING CHANGES or VIRTUAL CHANGES).

The new Change Management brings surrounding functionality into the picture and allows users to better make the change and propagate it into other environments. CM brings all the required change steps under one umbrella. In a manner that allows you to work on changes cooperatively, also on the same database object(s). The structured way is enforced by using a core set of processes that lead to the implementation of the change.

The processes are backed by a CM database that reflects the overall change situation within that DB2 environment (on a given DB2 subsystem).

When working with CM, you must make also a decision about the usage strategy of CM. One of three operating modes can be chosen as default:

- Mandatory:

  Enforce CM for all users for every change done in DB2 Admin and Compare.
- Optional:

  Get prompted for every change request whether CM shall be used or not.
- None:

  Do not use CM at all (work without CM for changes).

This setting is on the product level and applies to every user. However, there is the possibility to be more granular at the user level. This allows further combinations to better fit customers stepping into CM or planning to partially use CM. The default CM settings can be determined as follows, starting from the DB2 Admin Main Menu shown in Figure 4-3.

```
DB2 Admin -------------- DB2 Administration Menu 7.2.0 ------------------ 14:42
Option ===> cm

  1 - DB2 system catalog                                DB2 System: DB8A
  2 - Execute SQL statements                            DB2 SQL ID: PAOLOR2
  3 - DB2 performance queries                           Userid    : PAOLOR2
  4 - Change current SQL ID                             DB2 Rel   : 815
  5 - Utility generation using LISTDEFs and TEMPLATEs
  P - Change DB2 Admin parameters
 DD - Distributed DB2 systems
  E - Explain
  Z - DB2 system administration
 SM - Space management functions
  W - Manage work statement lists
  X - Exit DB2 Admin

 CM - Change management
                                                                   More:     +
Interface to other DB2 products and offerings:
   I  DB2I
   C  DB2 Object Comparison Tool
```

Figure 4-3 CM is active as shown on the main menu and can be selected

Attention: If there is no option CM on the main menu, then you can conclude that Change Management has not been activated. See also 2.1.7, “Enabling Change Management in your ISPF table” on page 23, where the **:CMOWN** parameter is described.

Next we choose the option **5** as shown in Figure 4-4.

```
DB2 Admin -------------------- Change Management (CM) ------------------- 14:38
Option ===> 5


   1 - Manage changes                                    DB2 System: DB8A
   2 - Manage masks                                      DB2 SQL ID: PAOLOR2
   3 - Manage ignores
   4 - Manage versions
   5 - Manage ID table
   6 - Report changes
```

Figure 4-4 Change Management main menu

And then we get to the default Change Management level, where we can see whether certain users hold individual options for the CM usage (see Figure 4-5).

```
DB2 Admin -------------- CM - Manage ID Table --------------------- Row 1 of 3
Command ===>                                              Scroll ===> PAGE

Commands: SAVE
Line commands:  D - Delete  I - Insert

Change management default level setting : NONE


Sel SQL ID   Level    Oper
    *        *        *
--- -------- -------- ------
    PAOLOR1  REQUIRED
    PAOLOR2  OPTIONAL
    PAOLOR3  OPTIONAL
****************************** END OF DB2 DATA ******************************
```

Figure 4-5 CM- Manage ID Table entries

In this example, the default setting is NONE (for all users there is no CM invocation at all when defining changes). This is the same mode as working without CM or as the product behaved before V720.

In addition, there are individual user entries (represented by SQLIDs) which can also be groups working with the same SQLID. In our case, for PAOLOR1, there is CM enforced for every change definition. For PAOLOR2 (and PAOLOR3) there is prompting for every change definition as to whether the change shall be registered in CM.

To test CM for certain users, set:

- One SQLID to either optional or required (like for PAOLOR2 and PAOLOR3)
- Change Management default level setting: NONE

To force CM for one application, set:

- SQLID for application DBA ID to REQUIRED (like for PAOLOR1)
- Change Management default level setting: NONE

4.2.2 A change consists of a set of SQL statements

Every change gets a unique change ID from the product (this cannot be influenced by the user) in ascending sequence. This change ID then ties together all change activity that belongs to it. Of course, you can give every change a more meaningful name so that naming conventions can be followed, or so that you can more easily remember what the change was about.

The CM - Changes panel shows all the changes in the CM database. Using this view, you an get an overview of the different status that changes go through (or the different status that changes can be in). See Figure 4-6.

```
DB2 Admin ------------------ CM - Changes ------------------------ Row 1 of 27
Command ===>                                                  Scroll ===> PAGE

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- -----------------> ------- -------- -------------------->
          28 PAOLOR2  C012                COMPARE COMPLETE PROMOTE DELTA V0-V1
st        27 PAOLOR2  C011                CHANGE  COMPLETE ALC DROP COLUMN
          26 PAOLOR2  C010                CHANGE  INITIAL  DROP COLUMN IN PAOLOR
          25 PAOLOR2  C009                CHANGE  DEFINED  COMPARE PAOLO2R PAOLO
          24 PAOLOR1  D070226-CR#10-NOR   CHANGE  DEFINED
          23 PAOLOR1  D070226-CR#09-NOR   CHANGE  DEFINED
          22 PAOLOR1  D070226-CR#08-NOR   CHANGE  CANCELED
          21 PAOLOR1  D070226-CR#07-IMM   FAST    COMPLETE
          19 PAOLOR2  DROP INDEX AGAIN    CHANGE  COMPLETE FROMPAOLOR2Z.GLWTACT
          18 PAOLOR2  ADD INDEX TO GLWTA  CHANGE  DEFINED  CHECK TRIGGER PROBLEM
          17 PAOLOR2  C008                CHANGE  CANCELED IMPORT PAOLOR2Z WITH
          16 PAOLOR2  C007                CHANGE  CANCELED IMPORT WITH TRIGGERS
          15 PAOLOR2  C006                CHANGE  CANCELED IMPORT WITH MASK
```

Figure 4-6 Display CM Changes panel, giving an overview of changes

Within the CM database, the change is stored as a set of SQL statements or pseudo SQL statements. These SQL statements reflect what has been defined as change (via AL, ALC, and so on) and then the product inserted the change statements via the REGISTER process. At this point we only have the definition of a change. Then later the ANALYZE process picks up these change definitions and generates a WSL containing the real actions (such as unload/drop/create/load).

To see an example, we selected change ID 27 and entered **ST** (to display the change statements). From there, we get the panel in Figure 4-7, showing all the change statements.

```
DB2 Admin ------------- CM - Change Statements -------------------- Row 1 of 1
Command ===>                                                   Scroll ===> PAGE

Change statements for change "PAOLOR2"."C011"
Line commands:
 E - Edit  D - Delete  I - Insert  S - Show

Sel    Sequence O  Qual     Name                Statement
              * *  *        *                   *
--- ----------- -- -------- ------------------ ------------------------------>

              1 TB PAOLOR2Z GLWTACT             ADMIN ALTER TABLE "PAOLOR2Z"."G
```

Figure 4-7 CM Change Statements panel showing all the change statements for a change

To expand on it, we then selected option **S** to browse the content. See Figure 4-8.

```
 Menu  Utilities  Compilers  Help
 SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
 BROWSE    SYS07058.T145928.RA000.PAOLOR2.R0137645    Line 00000000 Col 001 080
 Command ===>                                                   Scroll ===> CSR
********************************** Top of Data **********************************
 ADMIN ALTER TABLE "PAOLOR2Z"."GLWTACT"    DROP COLUMN "ACTKWD_NEW"
******************************** Bottom of Data *********************************
```

Figure 4-8 Expanding the complete statement in browse mode

So we actually see the statement. In this case it is not a valid (executable) SQL statement. The syntax to drop a column does not yet exist in DB2. However, this statement is prefixed with the ADMIN command, which is in an internal language that DB2 Admin understands, and it can transform this statement into a real change action. That change action would then be generated in the ANALYZE process, resulting in a work statement list (WSL) with all the required steps to accomplish this change request (unload data/drop/create/load etc.). Statements that are *registered* merely indicate “what” has to be done, rather “how” it is done; this is the function of ANALYZE.

4.2.3 Generating changes

Changes that can be registered and managed through Change Management can be generated either through the functions existing in DB2 Admin before V72 or new functions that have been introduced with Admin V72.

See Figure 4-9 for a summary of the three options available.

REGISTER process

The already existing DB2 Admin and Compare functions (if CM is activated for the user defining a change) are activated through the REGISTER process. The change definition commands that trigger the REGISTER process are:

- ALTER (AL, ALC, AL/RDEF, AL/RENAME)
- COMPARE (through DB2 Object Compare)
- CREATE/DROP in DB2 Admin

Import from a DDL file

A new function with CM to import change definitions is the Import from a DDL file:

- Delta changes data set coming from a PROMOTE
- Delta changes data set coming from a COMPARE
- DDL data set (coming from any source)

ISPF interface

There is also a new function with CM that allows you to generate changes through an ISPF interface. You can:

- CREATE the change (meaning register it)
- INSERT changes via **CS** (create changes) line command

Besides inserting, updating, and deleting change statements, it is also possible to generatea change via ISPF panels. However, users should be aware that the REGISTER process is not invoked; the change is created manually. When completed, register or re-register takes place.

Editing via this interface is considered as manual intervention. In this case, the prereqs are not re-established. It is the responsibility of the user to place this change in the prereq hierarchy by manipulating the prereq entries. Failing to do so can cause Merge and Run time failures.

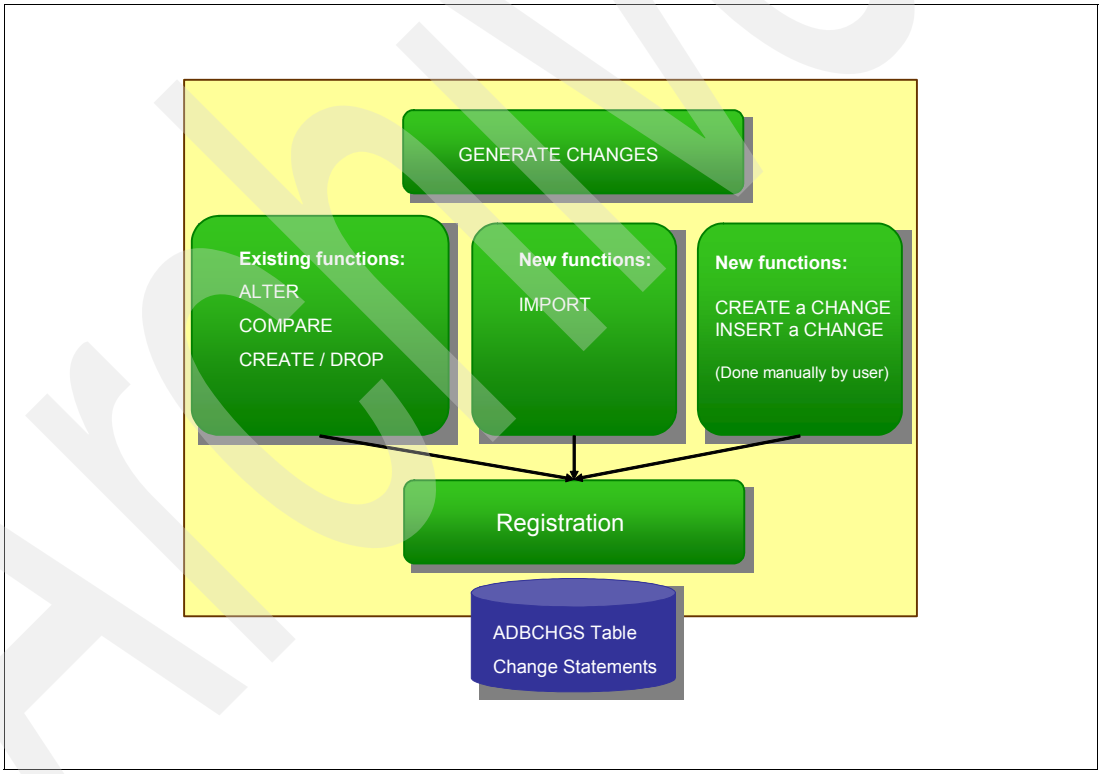

Figure 4-9 Overview on generating changes.

4.2.4 Set of changes can be promoted to other systems and imported there

A noticeable area of enhancements to Admin is to create a DELTA file that consists of only the specific changes between two different versions. This DELTA file can then can be promoted to other systems. The DELTA could be created on a development system and promoted to a test or production system. The receiving system is the target environment.

The receiving site of such *delta changes* is doing an IMPORT of the DELTA. The delta changes data set contains just the planned changes and as such gets registered in the CM database. In other words, the IMPORT process is nothing else than a REGISTER process.

The big difference is that the IMPORT allows you to change names in the delta changes data set (like database name, table space name, table name, etc.) to reflect the naming conventions at the receiving target side. This is facilitated through MASKING, which was already available in the product (see Figure 4-10).

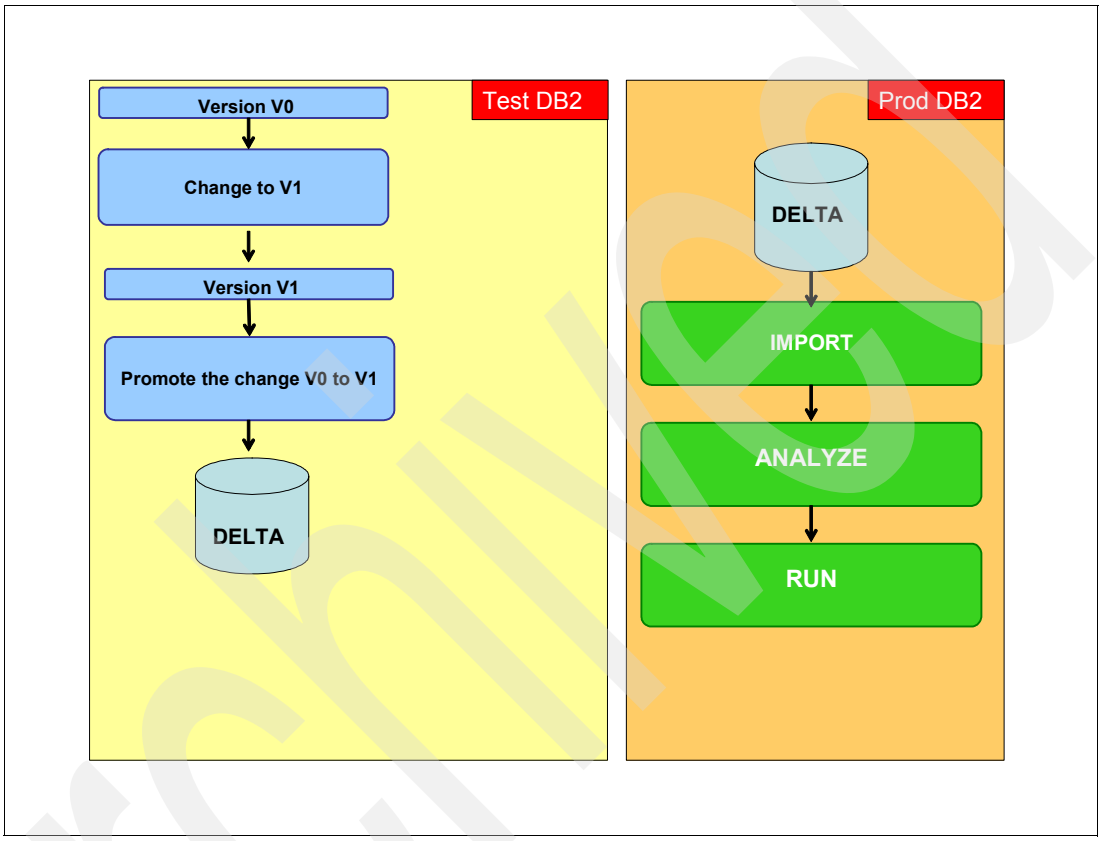

Figure 4-10 Overview of process flow to promote a change into another environment

The changes coming from the delta changes data set get *registered* through the IMPORT process and then they follow the normal flow with ANALYZE and RUN:

- IMPORT
- ANALYZE
- RUN

The changes can be imported repeatedly on different target environments. The IMPORTs are all based on the same delta changes data set, and you only have to specify appropriate masking so that after each IMPORT, the changes are registered with the naming conventions of the target environment.

The delta changes data sets are generated through comparison of Version Files that constitute the before and after image of a change. This even allows to generate delta changes data sets that span multiple versions (starting version V0, ending version Vn) in case one version requires more than one change incorporated in the delta changes data set.

The DELTA is shown in Figure 4-11.

```
ALTER     ADMIN ALTER TABLE PAOLOR2Z.GLWTACT
ALTER       DROP COLUMN ACTKWD_NEW;
```

Figure 4-11 The content of a delta changes data set

This delta changes data set contains the description of the change and as such, gets registered with the IMPORT. The change statements are then recorded in the CM database.

4.2.5 Changes are stored in a CM data base (one per subsystem)

All changes that go through CM are stored in the CM database. If changes are made outside of CM — which depends on how CM is set up — then these changes are NOT reflected in the CM database. One can see this CM database as repository of all change requests for a given DB2 subsystem.

Every CM database consists of its own set of tables (see Table 4-1).

Table 4-1 CM data base tables

Table name	Contents
ADBCHG	Parent table for all changes, change-id is an identity column, change-id is primary key for other Change Management tables.
ADBCHGS	The changes table contains all the change statements by change-id. There is one row per statement. If the statement is more than 3000 chars, it spans across multiple rows.
ADBCHGSR	Related objects table contains the related objects of that change (change on table shows table, table space, and database).
ADBCID	Change Management user ID table, contains the default level of Change Management (NONE, OPTIONAL or REQUIRED) and the level for individual SQLIDs.
ADBCIGNORE	IGNORE parent table, one entry per ignore name.
ADBCIGNORES	IGNORE child table, multiple entries per ignore name.
ADBCMASK	MASK parent table, one entry per mask name.
ADBCMASKS	MASK child table, multiple entries per mask name.
ADBCPREREQ	Prerequisites table, entries are added when registering a new change and there are existing changes which are identified as prereqs for this change.
ADBCVERLINES	Version child table, contains multiple entries per version ID.
ADBCVERSCOPE	Scope parent table, has one entry per scope ID identity column.
ADBCVERSCOPES	Scope child table, has multiple entries per scope ID.
ADBCVERSION	Version parent table (child is ADBCVERLINES table), contains one row per version name, version ID is identity column and primary key. If it is a base overstepped, it also points to an ADBCSCOPE.

The most important table and the starting point (or anchor point) is the ADBCHG table. It is the root table for all changes. The CHANGEID column itself is an identity column, and for every new change, there is a new CHANGEID assigned from DB2.

This ADBCHG table ties together all information about a given change, and you can very easily get an overview of the status of the change. The diagram in Figure 4-12 shows you how the tables are linked together.

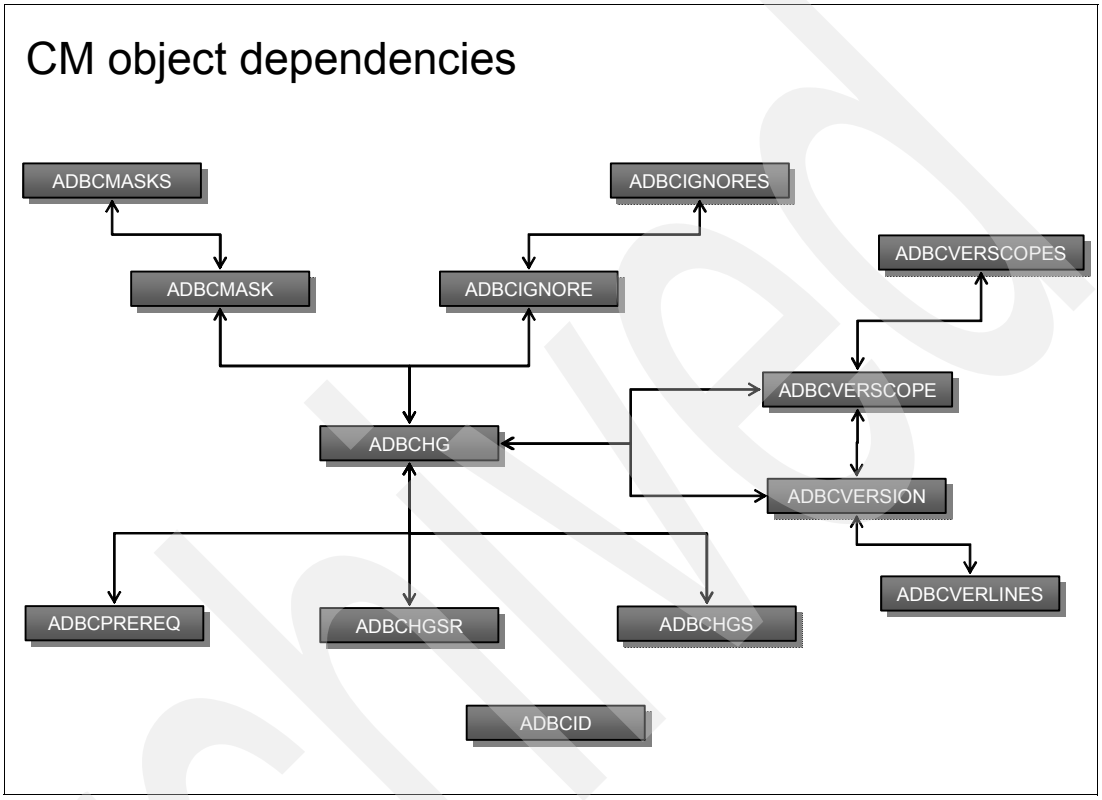

Figure 4-12 Tables relationships

In a multiple DB2 environment, there might arise the requirement to view the status of a change across all the DB2s where the change has been rolled out — for example, whether a change made in the development environment has already been implemented in production.

Currently this view is not possible. One has to go to every DB2 and check the status of a specific change. In Figure 4-13 we show this independency between CM databases.

Archived

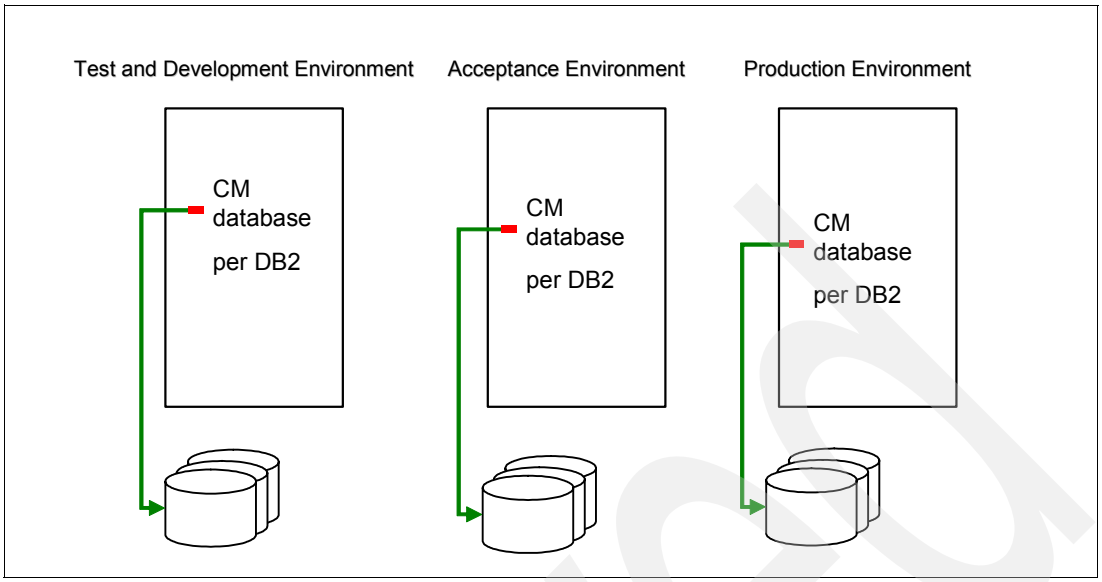

Figure 4-13 Overview of CM databases in multiple DB2 environments.

Important: Change requests from other DB2 environments cannot be seen. There is no link between these CM databases and no communication telling you that a change has been rolled out into all receiving target environments.

Even when CM forces all changes (made within DB2 Admin and DB2 Object Compare) to go into the database, there is still the possibility that users can work externally (that is outside DB2 Admin and DB2 Object Compare) to make a change, like using SPUFI or any other means. Such changes (external to DB2 Admin and DB2 Object Compare) are not reflected in the CM database. This means that there must be a strong emphasis being put on establishing and keeping the change procedures defined within a given environment.

There is the possibility for DB2 Admin to have more than one CM database. This might make sense for testing purposes or when you want to isolate changes. But we expect this to be a rarer case.

4.2.6 Answering questions such as “Who did what, when?”

Before CM, there was no way to find out about a change that was done in pre-CM days. With a CM database, one gets a view into the changes itself, the status of a change, who works on the change, and which objects are involved in the change. CM goes far beyond that because it allows you to dig into all facets of a change which are involved in the change (which version files were used, which prerequisites exist for the change, which version scope was used, etc.) Every contributing factor that led to the change is stored in the CM database and can be queried.

Figure 4-14 is a coding example that displays changes in general.

```
ADB2C11 n ------------------ CM - Changes ------------------------ Row 1 of 30
Command ===>                                                  Scroll ===> PAGE

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- -----------------> ------- -------- -------------------->
          31 PAOLOR2  R014                RECOVER COMPLETE RECOVER CHANGE FOR CH
          30 PAOLOR2  C014                CHANGE  DEFINED  IMPORT STMT TAKEN FRO
          29 PAOLOR2  C013                CHANGE  INITIAL  CREATED IN ISPF
          28 PAOLOR2  C012                COMPARE COMPLETE PROMOTE DELTA V0-V1
          27 PAOLOR2  C011                CHANGE  COMPLETE ALC DROP COLUMN
          26 PAOLOR2  C010                CHANGE  INITIAL  DROP COLUMN IN PAOLOR
          25 PAOLOR2  C009                CHANGE  CANCELED COMPARE PAOLO2R PAOLO
          24 PAOLOR1  D070226-CR#10-NOR   CHANGE  DEFINED
          23 PAOLOR1  D070226-CR#09-NOR   CHANGE  DEFINED
          22 PAOLOR1  D070226-CR#08-NOR   CHANGE  CANCELED
          21 PAOLOR1  D070226-CR#07-IMM   FAST    COMPLETE
          19 PAOLOR2  DROP INDEX AGAIN    CHANGE  COMPLETE FROMPAOLOR2Z.GLWTACT
          18 PAOLOR2  ADD INDEX TO GLWTA  CHANGE  DEFINED  CHECK TRIGGER PROBLEM
```

Figure 4-14 Example of CM-Changes

Figure 4-15 is another coding example that displays changes from an object perspective.

```
ADB2C62 n -------------- CM - Changed Objects ------------------- Row 1 of 373
Command ===>                                                  Scroll ===> PAGE

Line commands:
 ST - Statements  CH - Change  CHA - All Changes  CHC - Completed Changes
 CHN - Not Completed Changes

         Change Change   Change              Object    Object
Sel    Sequence Owner    Name             O  Qualifier Name
               * *        *                *  *         *
--- ----------- -------- ---------------- -- --------- ------------------
              1 PAOLOR1  D070221-CR#01-I   TB DSN8810   DEPT
              1 PAOLOR1  D070221-CR#02-N   TB DSN8810   EDEPT
              1 PAOLOR1  D070222-CR#03-IMM SG           DGDBSTOG
              1 PAOLOR1  D070222-CR#04-CHG TB F419838   TBE015A1
              1 PAOLOR2  C003              TB PAOLOR2Z  GLWTACT
              1 PAOLOR2  C004              DB           PAOLOR2
              2 PAOLOR2  C004              TS PAOLOR2   GLWS001
              3 PAOLOR2  C004              TB PAOLOR2   GLWTACT
              4 PAOLOR2  C004              TB PAOLOR2   GLWTACT
              5 PAOLOR2  C004              IX PAOLOR2   GLWXACT1
              6 PAOLOR2  C004              TB PAOLOR2   GLWTJBS
              7 PAOLOR2  C004              TB PAOLOR2   GLWTJBS
              8 PAOLOR2  C004              IX PAOLOR2   GLWXJBS1
```

Figure 4-15 CM Changed Objects panel - All affected objects for a change

After gaining knowledge and experience, you might choose to directly access the underlying tables of the CM database.

4.2.7 Immediate deployment possible for small changes

As with all rules and processes, there can be exceptions to deviate from the standard process (REGISTER, ANALYZE, RUN). This applies to changes that can be executed instantly without going through all these steps, which are normally required.

Such fast changes can be executed immediately. If an immediate or emergency change is applied to an object that has been defined within the scope of Change Management, caution is required, since once such a change is executed, the recover capability is lost for the changes until then.

Fast changes are all of the same type (ALTER, CREATE, DROP) in the sense that they can be processed through DB2 directly via SQL statement. Within DB2 Admin there are different ways in which the product recognizes an immediate change:

- AL function (against table, index, table space, database) which results in an SQL ALTER
- Submitting SQL statements directly through DB2 Admin (ALTER/CREATE/DROP) also triggers an immediate change

Changes that are made using the following methods cannot be run as immediate changes:

- ALC
- AL/RDEF
- AL/RENAME
- COMPARE

These changes can be destructive in nature (leading into an unload/drop/create/load) and of course cannot be run immediately. Even if we specify changes through these functions (as in ALC, adding a column to the end of the table) which could be run immediately, DB2 Admin takes the change request through the normal flow of actions, that is ANALYZE and RUN.

And as with any other change, the immediate change is also subject to pending changes. If there are any changes defined (and not yet done), then we have to decide what to do:

- Cancel (the immediate change)
- Register and run as an emergency change
- Register as a normal change, pending changes become prereqs
- Register as a normal change, supersede pending changes

If you want to know more about this topic, refer to 4.4, “Change processes” on page 133. There we explain in detail pending changes, immediate changes, emergency changes, and other related issues.

4.2.8 Changes can be recovered

With every change that occurs, there is a chance that a completed change might have to be recovered — meaning that the structure should be reinstated as it was before the change.

This can occur if a change was successfully implemented and made available: On Monday morning the first applications run against objects which have been subject to that change and all of a sudden there is a problem with the new structure. Either some applications were not fully tested or adapting the new structure, or the change itself seems to cause performance degradation, or any other malfunction is assumed to originate with the completed change.

Now you have the option to revert that change back to the state it was before.

The RECOVER change then has all the actions in place (WSL is generated and ready to be executed) to revert the change. All it requires later for the RECOVER action is to generate the JCL for the RECOVER WSL and run it. See Figure 4-16.

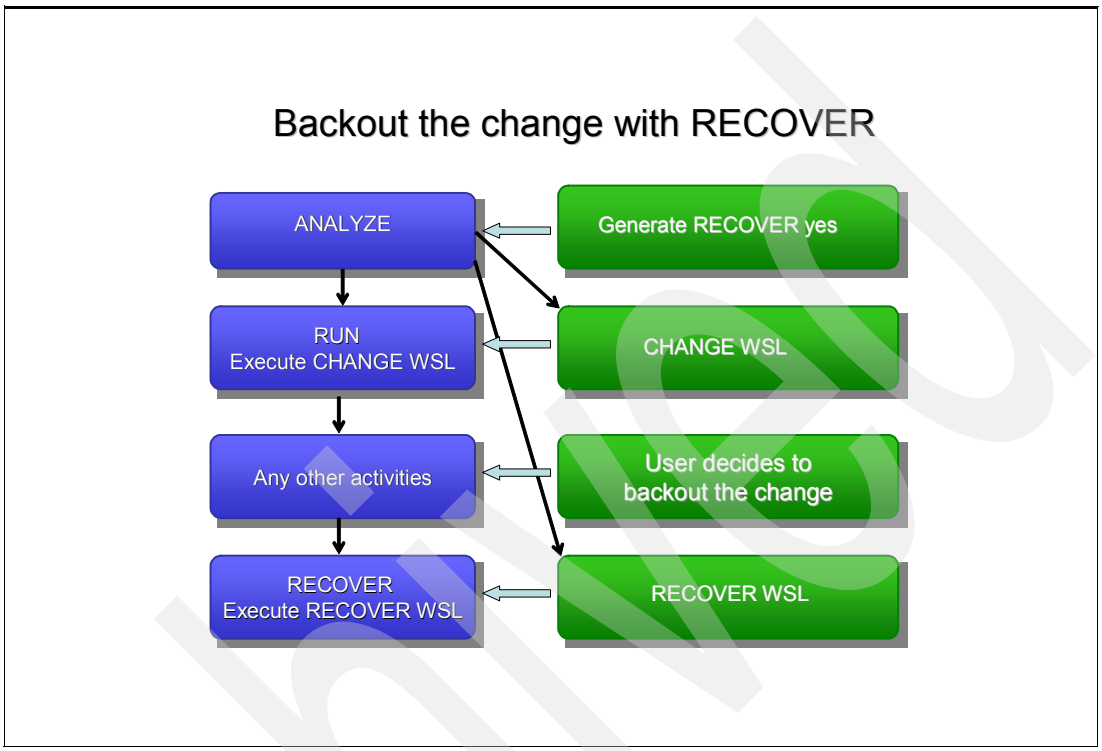

Figure 4-16 Change backout with RECOVER

Important: The RECOVER function works on COMPLETED changes.

Another important point to remember is that a RECOVER WSL spans always one version. This is because RECOVER WSL gets generated at ANALYZE time, so it can only cover one version change.

If you have to back out multiple versions (such as from V4 to V1) then you have to revert back version for version starting from current (V4 to V3, V3 to V2, V2 to V1). So RECOVER cannot be used to recover V4 to V1 in one step because there is no such WSL supporting this.

Archived

The RECOVER process is illustrated in Figure 4-17.

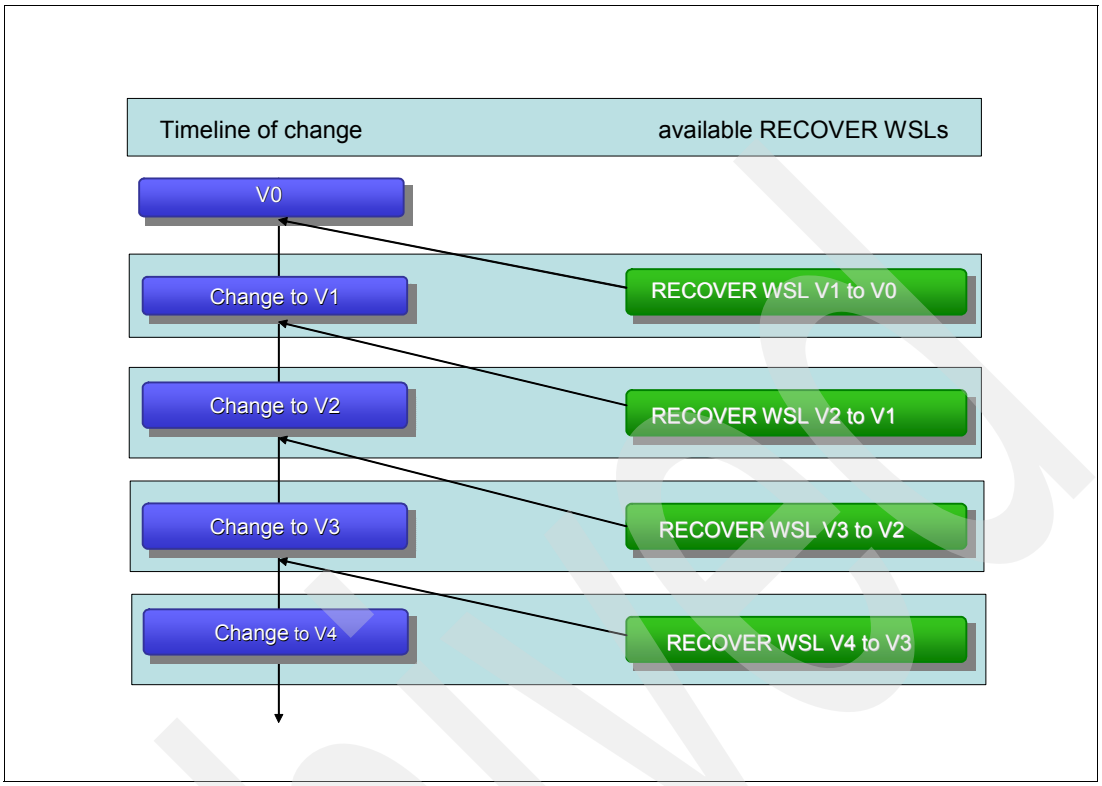

Figure 4-17 Granularity of RECOVER

So this should be enough to get you started and give you a basic idea of how it all works. Now let us dive into the details of the new CM functionality to understand terminology, processes, change types, and so on.

4.3 Change Navigation

Assuming that your product setup is done as described in Chapter 2, “Customization” on page 15, you can invoke Change Management from the Admin Main Menu by typing **CM**. The upcoming menu is called the Change Management Menu. See Figure 4-18.

```
DB2 Admin -------------------- Change Management (CM) ------------------- 11:04
Option ===>


   1 - Manage changes                                    DB2 System: DB8A
   2 - Manage masks                                      DB2 SQL ID: PAOLOR1
   3 - Manage ignores
   4 - Manage versions
   5 - Manage ID table
   6 - Report changes
```

Figure 4-18 Change Management Menu

4.3.1 Looking up a change

You can query your changes starting from the Change Management menu. You can invoke the Change Management menu from the Admin Main Menu with the command **CM** or from wherever you are within Admin with the command **CMM**.

Invoke from the Change Management Menu using option **1** - Manage changes. Depending on your search criteria, you can display a subgroup of your changes. See Figure 4-19.

```
DB2 Admin ---------------------  CM - Manage Changes -------------------- 11:18
Option ===> 1


   1 - Display changes                                   DB2 System: DB8A
   2 - Create a change                                   DB2 SQL ID: PAOLOR1
   3 - Create delta for target
   4 - Import changes









Enter display selection criteria (Using a LIKE operator, criteria saved):
Name . . . . . . .                    >  Created by . . PAOLOR1% >
Owner  . . . . . .          >            Altered by . .          >
Type . . . . . . .                       Status . . . . ANALYZED
Created within . . 6 DAYS                Change ID  . .
Altered within . .
```

Figure 4-19 Query your changes

In this example, we did a search of all changes in Change Management which have been entered within the last 5 days with its status ANALYZED created from the userid PAOLOR1.

The result is the Changes summary panel shown in Figure 4-20.

Archived

```
DB2 Admin ------------------ CM - Changes ------------------------- Row 1 of 3
Command ===>                                                  Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- --------------------
         202 PAOLOR1  D070313-CR#55-NOR  CHANGE  ANALYZED
         220 PAOLOR1  D070315-CR#60-NOR  CHANGE  ANALYZED
         231 PAOLOR1  D070315-CR#63-NOR  CHANGE  ANALYZED
****************************** END OF DB2 DATA ******************************
```

Figure 4-20 The Changes summary panel

4.3.2 Available commands from the change summary panel

From the change summary panel, you can enter various commands in front of every change. They are listed in Table 4-2. Note that some of them are not listed on the Change summary panel. You can enter **?** as a line command to see all the available commands. We also provide a link where you can find related information not just to the command but to the connected topic itself.

Table 4-2 List of available commands for change

Command	Description	References to further information
AN	Analyze a change	4.4.3, “The ANALYZE process” on page 139
CAN	Cancel a change	4.7, “Canceling a change” on page 164
CP	Create a prerequisite	4.11.7, “Prerequisite change” on page 185
CS	Create a change statement	, “REGISTER of a normal change” on page 135
EA	Edit the analyze job	4.4.3, “The ANALYZE process” on page 139
ER	Edit the run job or promote job	“Restarting a RUN job” on page 148 and 4.4.5, “The PROMOTE process” on page 151
I	Details on change	
IG	Show ignores	
IM	Import DDL	
INS	Insert a change	
MA	Show masks	
OR	Show original change	
PQ	Show prerequisites	4.11.7, “Prerequisite change” on page 185
PR	Promote a change	4.4.5, “The PROMOTE process” on page 151
RC	Recover a change	4.4.7, “The RECOVER process” on page 158

Command	Description	References to further information
RE	Show recover change	
RN	Run a change	4.4.4, “The RUN process” on page 145
RST	Restart the change	4.8, “Restarting” on page 164
SC	Show version scopes	
ST	Show change statements	4.3.3, “Showing change statements” on page 131
U	Update a change	
V	Validate WSL	4.4.3, “The ANALYZE process” on page 139
VE	Show versions	

4.3.3 Showing change statements

Like the SR command, you can use the **ST** line command to display the change statements related to a change, as shown in Figure 4-21.

```
DB2 Admin ------------------ CM - Changes ------------------------ Row 4 of 4
Command ===>                                                  Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                 Type    Status   Comment
           * *        *                    *       *        *
--- -------- -------- ------------------ ------- -------- ---------------------
ST       231 PAOLOR1  D070315-CR#63-NOR  CHANGE  ANALYZED
****************************** END OF DB2 DATA ******************************
```

Figure 4-21 Show statements of a change

You are presented with the statements belonging to your change. See in Figure 4-22.

```
DB2 Admin ------------- CM - Change Statements -------------------- Row 1 of 1
Command ===>                                                  Scroll ===> CSR

Change statements for change "PAOLOR1"."D070315-CR#63-NOR"
Line commands:
 E - Edit  D - Delete  I - Insert  S - Show

Sel    Sequence O  Qual     Name                 Statement
              * *  *        *                    *
--- ----------- -- -------- ------------------ ------------------------------>

              1 TS DSN8D81A DSN8S81S           ALTER TABLESPACE DSN8D81A.DSN8S8
****************************** END OF DB2 DATA ******************************
```

Figure 4-22 The statement summary

There are three ways to see the statements:

- Browse all the table contents. You can use PF10 / PF11, then to go to the left and to the right. Enter **BR** as a primary command.
- Show an individual statement. Enter **S** as a line command in front of each statement.
- Position your cursor elsewhere in the statement. According to the example in Figure 4-22, position your cursor on any character of the statement ALTER TABLESPACE. In order to see the rest of the statement, you can use the PF10 and PF11 keys to switch to the left and to the right.

4.3.4 Refreshing your changes

You can enter the command **REF** to refresh your change panel. This is necessary, for instance, if you perform an action against a change like RUN or ANALYZE. The status is usually not automatically updated when you go back to the change overview panel. Therefore you should always enter the **REF** command to see the new status after doing actions against your changes.

4.3.5 REPORT changes

Admin also provides a possibility to report on generated changes. For instance, you can search for objects created before or after a given date. The most important option of the REPORT facility is that you can show the affected changes related to a specific object. See Figure 4-23.

```
DB2 Admin ------------------ CM - Report Changes ---------------------- 18:29
Option ===> 2


   1 - Display changes                                DB2 System: DB8A
   2 - Display changed objects                        DB2 SQL ID: PAOLOR1









Enter display selection criteria (Using a LIKE operator, criteria saved):
Name . . . . . . . DSN8%              >   Created by . . . .          >
Owner  . . . . . .          >             Altered by . . . .          >
Type . . . . . . . TS                     Status . . . . . .
Created before . .                        Altered before . .
Created after  . .                        Altered after  . .
```

Figure 4-23 Report from changes

According to this example, you get a list of all changes that affected a table space (type TS) starting with DSN8.

Depending on your action: **1** - Display changes or **2** - Display changed objects — notice that name and owner are used in two different manners in Figure 4-24. For option **1**, name and owner are the change name and change owner. For option **2**, name and owner are the DB2 object names, depending on your type entered.

```
DB2 Admin -------------- CM - Changed Objects --------------------- Row 1 of 7
Command ===>                                                  Scroll ===> CSR

Line commands:
 ST - Statements  CH - Change  CHA - All Changes  CHC - Completed Changes
 CHN - Not Completed Changes

         Change Change   Change                Object    Object
Sel    Sequence Owner    Name              O  Qualifier Name
              * *        *                 *  *         *
--- ----------- -------- ----------------- -- --------- -----------------
              1 PAOLOR1  D070313-CR#55-NOR TS DSN8D81A  DSN8S81S
              2 PAOLOR1  D070313-CR#55-NOR TS DSN8D81A  DSN8S81S
              1 PAOLOR1  D070315-CR#57-NOR TS DSN8D81A  DSN8S81S
              2 PAOLOR1  D070315-CR#58-NOR TS DSN8D81A  DSN8S81S
              1 PAOLOR1  D070315-CR#59-NOR TS DSN8D81A  DSN8S81S
              2 PAOLOR1  D070315-CR#60-NOR TS DSN8D81A  DSN8S81S
              1 PAOLOR1  D070315-CR#63-NOR TS DSN8D81A  DSN8S81S
****************************** END OF DB2 DATA ******************************
```

Figure 4-24 Looking up changed objects

You can look up changes for the following types of DB2 objects:

- AL - Alias
- AT - Auxiliary table
- DB - Database
- DT - Distinct type
- FU - Function
- GT - Global temporary table
- IX - Index
- MQ - MQT
- SQ - Sequence
- SG - Storage group
- SP - Stored procedure
- TB - Table
- TS - Table space
- TR - Trigger
- SY - Synonym
- VW - View

4.4 Change processes

The following diagrams and descriptions should give you a basic understanding about all the processes which are currently being covered by Change Management. Notice that every change process results in a specific status of your change entered. There is a description of states at a glance in Table 4-5 on page 162. In the end of every process description, we provide a brief summary of all the states that are considered to be for each process.

Archived

Table 4-3 shows an overview of all the processes involved in Change Management.

Table 4-3 Processes and link to description

Process	Link to where it is described
REGISTER	“The REGISTER process” on page 134
ANALYZE	4.4.3, “The ANALYZE process” on page 139
RUN	4.4.4, “The RUN process” on page 145
PROMOTE	4.4.5, “The PROMOTE process” on page 151
IMPORT	4.4.6, “The IMPORT process” on page 155
RECOVER	4.4.7, “The RECOVER process” on page 158

4.4.1 Status of a change

Before talking about the different processes we want to give you a brief overview of the different states a change can have. You are confronted with two different categories of changes you can currently create with Change Management. See Figure 4-25.

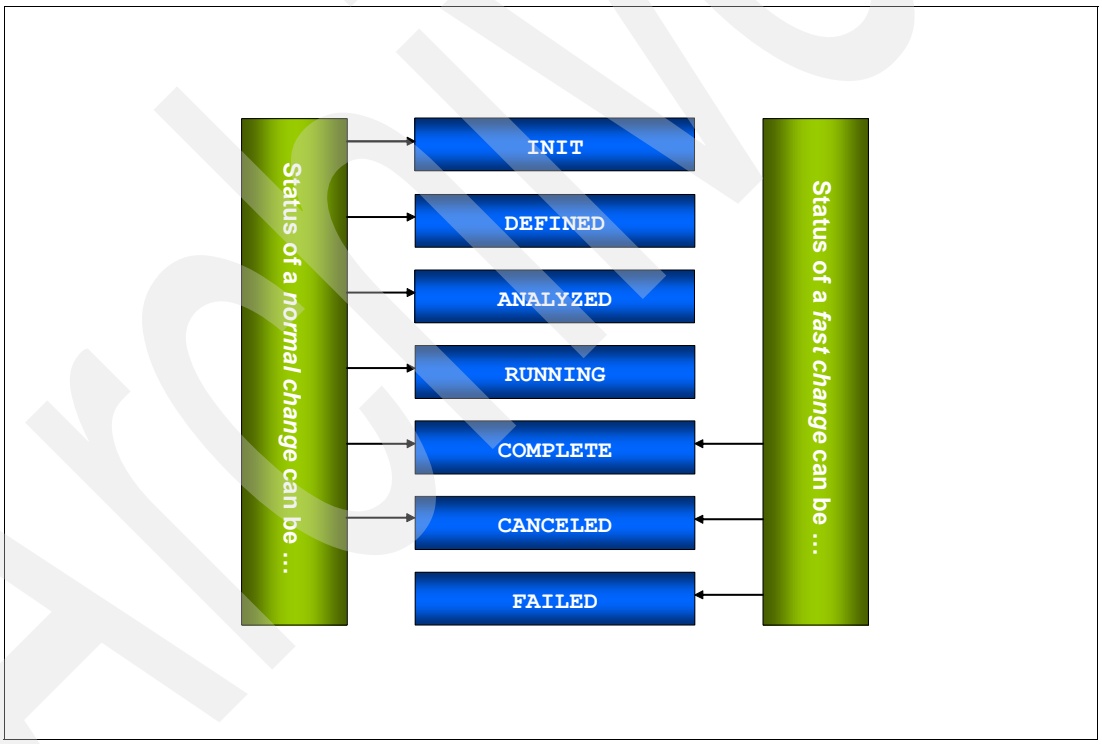

Figure 4-25 Summary of states

4.4.2 The REGISTER process

In this section we discuss the REGISTER process of a normal change and a fast change. The REGISTER process itself always follows the workflow described in Figure 4-26.

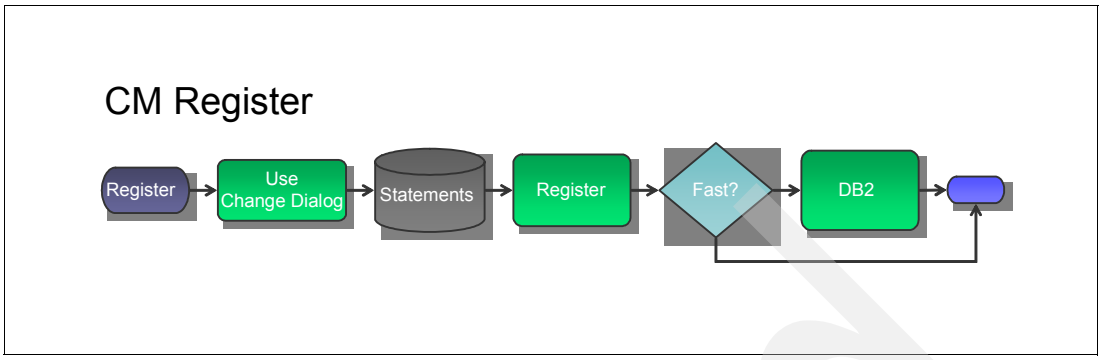

Figure 4-26 The REGISTER process

REGISTER of a normal change

First of all, you have to define a change. A simple ALC on a table, that is, just adding or altering a column, is already a definition. At the time of a change definition, there is nothing inside the Change Management database.

You then do a Register of our defined change. A change becomes registered as soon as it is inserted in the Change Management database. When you try to look up the change with the option *Manage changes* and then *Display changes* from the Change Management Menu, you can see that the status of your recently registered change is DEFINED. Therefore, definition and registration of a change go side by side.

Finally, registering the change in Change Management is inserting a row in table <:cmown>.ADBCHG.

A change can only be registered as DEFINED if there is at least one statement that belongs to a change. If no statement belongs to a change, then your change status becomes INIT. The INIT status appears, for instance, if you try to create a change from the Manage Changes Menu. Even in that case, your change becomes registered.

As a change is being registered in the Change Management database, a unique change ID is assigned to the change. The change IDs are based on identity columns and are used by DB2 Admin for internal reference within Change Management. See Figure 4-27.

Archived

```
DB2 Admin ---------------------  CM - Manage Changes -------------------- 20:19
Option ===> 2


   1 - Display changes                                    DB2 System: DB8A
   2 - Create a change                                    DB2 SQL ID: PAOLOR1
   3 - Create delta for target
   4 - Import changes









Enter display selection criteria (Using a LIKE operator, criteria saved):
Name . . . . . . .                    >  Created by . .          >
Owner  . . . . . .         >              Altered by . .          >
Type . . . . . . .                        Status . . . .
Created within . .                        Change ID  . .
Altered within . .
```

Figure 4-27 Create a change with status INIT

You can provide the required information for owner and name and your change becomes registered with its status INIT. See Figure 4-28.

```
DB2 Admin --------------- CM - Insert a Change ------------------------- 20:21
Command ===>
INSERT stmt executed
Enter the change information below:
Owner  . . . . . . . PAOLOR1  >  (Optional, default is PAOLOR1)
Name . . . . . . . . D070313-CR#55-NOR  >  (Required, ? to look up)
Comment  . . . . . .                                          >

Specify owner and name values to use for this change:
                    Owner        Name
Version scope  . . .         >                     >  (? to look up)
Source version . . .         >                     >  (? to look up)
Target version . . .         >                     >  (? to look up)
Base version . . . .         >                     >  (? to look up)
```

Figure 4-28 Inserting (or creating) a change

Tip: You should consider setting up a naming convention for your changes. This can be the date when the change has been entered or it can be the date your change is applied to production. You should also think about an abbreviation of the application area your change belongs to.

When you go back to the Change Menu, you find out that your change PAOLOR1.D070313-CR#55-NOR is in status INIT. This is, as mentioned above, because there is no statement currently referring to this change. See Figure 4-29.

```
DB2 Admin ------------------ CM - Changes ------------------------- Row 9 of 9
Command ===>                                                   Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- ---------------------
         202 PAOLOR1  D070313-CR#55-NOR  CHANGE  INITIAL
****************************** END OF DB2 DATA ******************************
```

Figure 4-29 Initial status of a change

There is a line command which offers you the opportunity to insert a statement to a change. Enter **CS** in front of this change and you are asked for a statement in edit mode of a temporary data set. Enter the statement of your choice to the temporary data set, for instance, an ALTER statement as shown in Example 4-1.

Example 4-1 ALTER statement represents your change

```
ALTER TABLESPACE DSN8D81A.DSN8S81S PRIQTY 48 SECQTY 48
```

PF3 triggers the restart automatically to invoke Register. Notice the special terminology here! Restarting a change using the command **RST** does mean to bring an INIT status into DEFINED status by doing a straight registry process.

There is also a line command, **RST**, which can be issued against a change if the change is in INITIAL status and must be restarted. This is intended for cases where you have not inserted, deleted, or updated change statements.

In the register panel, you are asked to complete your change and add a delta version owner and name. You can use the same names you used already for the change owner and name. See Figure 4-30.

Archived

```
DB2 Admin -------------------- CM - Register Options -------------------- 20:35
Option ===>
Enter required field
Commands: CONTINUE                                        DB2 System: DB8A
                                                          DB2 SQL ID: PAOLOR1
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR1  >  (Optional, default is PAOLOR1)
Name  . . . . . . . . D070313-CR#55-NOR                                    >
Comment . . . . . . .                                                      >
Change Type . . . . : CHANGE   (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                      Owner         Name
Ignore  . . . . . . .         >                          >  (? to look up)
Mask  . . . . . . . .         >                          >  (? to look up)
Delta Version . . . . PAOLOR1  >    D070313-CR#55-NOR    >  (? to look up)
```

Figure 4-30 Complete the change definition

Complete your change with CONTINUE, and you go directly to the change overview panel. You can see that your change status has been changed to DEFINED rather than INIT.

You have the ability to update your change with more than one statement. Enter the command ‘CS’ again and follow the process described above. Once you start editing the change statements, the change goes back to INITIAL status.

When REGISTER processes the statements successfully, the status is updated to DEFINED.

If you delete all statements from a change, after deleting the last change your change goes from DEFINE back to INIT status.

Enter the primary command ‘REF’ to refresh your change panel in order to see the actually change states.

Delta version

A delta version is a DB2 catalog representation of your change and is being created as soon as your change becomes registered.

REGISTER of a fast change

A fast change is an emergency change or an immediate change. The difference in registration between a fast and a normal change is that as soon as a fast change becomes registered, it is supposed to be executed. In case of a fast change, no delta version is created.

Fast changes are described in detail when we talk about change terminology. Refer to 4.11.5, “Emergency change” on page 180 and 4.11.6, “Immediate change” on page 184.

REGISTER using Compare

If you compare two environments using Compare, this can be a change registration as well. Your change automatically becomes registered as DEFINED inside Change Management. As with the usual Compare, you can use masking and ignore. Note that if you plan to use Compare with Change Management, there is a last step called the register step of your change. You can register a change just one time, for that reason, there is no restart of your Compare job possible.

Compare is based on comparing version files after creating them, depending on your input. You now have the option to keep these version files in Change Management. This is for users who wish to keep things in CM rather than manage data sets.

If you choose to store the version files in CM, enter an owner and a name for internally stored version files.

In order to use Change Management with DB2 Compare, make sure that you set the option GENERATE APPLY JOBS to YES when you enter the Compare job parameters. This is important because only then are you asked to use Change Management.

Notice that Compare does not produce base versions. Versions created during Compare are delta versions.

Compare also results in providing a delta change data set. You can use this data set to deploy your change to other DB2 environments. In these other DB2 environments you are asked to import this delta change data set. Refer to 4.4.6, “The IMPORT process” on page 155 to learn how to use the IMPORT process.

Refer also to Chapter 7, “Rolling out new versions of a database design” on page 311. There is a practical example given that shows you the way to use Compare.

Notice that the REGISTER process marks your change either in status INIT or DEFINED with the following exceptions:

- An immediate or fast changes goes directly to COMPLETE or FAILED.
- A PROMOTE change (to generate delta changes file) goes directly to ANALYZED.
- A recover change goes directly to ANALYZED.

4.4.3 The ANALYZE process

In this section we discuss the ANALYZE process. The ANALYZE process itself always follows the workflow described in Figure 4-31.

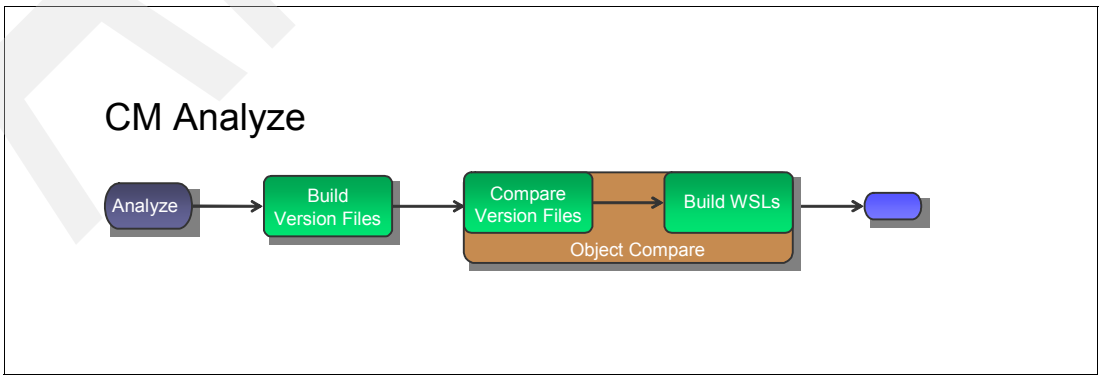

Figure 4-31 The ANALYZE workflow

After your normal change has been registered, you have to analyze this change. Within ANALYZE, your definitions are checked against the DB2 catalog. For instance, if you decided to append a column to a table, the ANALYZE process checks that the column to be appended has a valid name and format and that the affected table is available. In the same way, ANALYZE checks if the changes already exist. The ANALYZE process includes checking for consistency with prerequisite changes as well.

The recover change is registered as part of ANALYZE.

You have to enter the line command **AN** in front of your change to create a batch job that does the ANALYZE process. See Figure 4-32.

```
DB2 Admin ------------------ CM - Changes ------------------------- Row 8 of 8
Command ===>                                                  Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- ---------------------
AN       212 PAOLOR1  D070315-CR#59-NOR  CHANGE  DEFINED
****************************** END OF DB2 DATA ******************************
```

Figure 4-32 Using the ANALYZE command

Archived

Next, in Figure 4-33, we show the Generate Analyze Job panel.

```
DB2 Admin ---------------- Generate Analyze Job  ---------------------- 15:42
Command ===>

 Specify the following for Analyze:
                                                                  More:     +
 Base version method . . . . . A        (Auto, User, or Existing)

 Change reporting options  . . NO       (Yes/No)

 Required data set information:
   PDS for WSL . . . . . . . 'ADMIN.CM.WSL'
   PDS for batch jobs  . . . 'ADMIN.CM.CNTL'
   Prefix for data sets  . . ADMIN

 Options:
   Run SQLID . . . . . . . .          (Blank, a SQLID, or <NONE>)
   Validate WSL  . . . . . . NO       (Yes/No)
   Use utility options . . . NO       (Yes/No)
   Generate templates  . . . NO       (Yes/No)
   Build JCL to run WSL  . . NO       (Yes/No)
   Generate a recover change NO       (Yes/No)
     Data to recover . . . .          (Original or Existing)


   Optional jobs after Reload or Alter:
     Run CHECK DATA  . . . . NO       (Yes/No)
     Take an image copy  . . N        (after: Reload/Alter/Both/None)
     Run REORG . . . . . . . N        (Mandatory, All relevant, None)
     Run RUNSTATS  . . . . . N        (after: Reload/Alter/Both/None)
     Run REBIND  . . . . . . NO       (Yes/No)

 BP - Change batch job parameters
```

Figure 4-33 The ANALYZE process parameter panel to generate ANALYZE JCL

Some of the fields in Figure 4-33 are important, but are not self-explanatory, so we discuss them in more detail:

- Base version method:

  The ANALYZE process uses a comparison process of versions to finally build a WSL to be run with Change Management. Versions are an internal representation of DDL, they can be stored within Change Management (versions in general are described in 4.12, “Versions” on page 186.). The ANALYZE process uses versions to do an internal compare between a target base version and a source base version. Target and source base versions are built during ANALYZE and used for Compare reasons in your ANALYZE job. They are just temporary and are not stored in Change Management.

Archived

You can choose how your target base version is built by using one of the following three base version methods:

– Auto:

  If you choose **Auto**, the objects related to your change are chosen from the delta version file. A delta version file has already been created during the REGISTER process of your change. These objects are then processed by the **GEN** command internally. This is your internal (and temporary) target base version.

– User:

  If you choose **User**, the objects related to your change are taken from the version scope. They are then processed by the **GEN** command internally. This is your internal (and temporary) target base version.

– Existing:

  If you choose **Existing**, your target version file is being extracted from a base version stored within Change Management. Using an existing base version from Change Management helps you to save processing time.

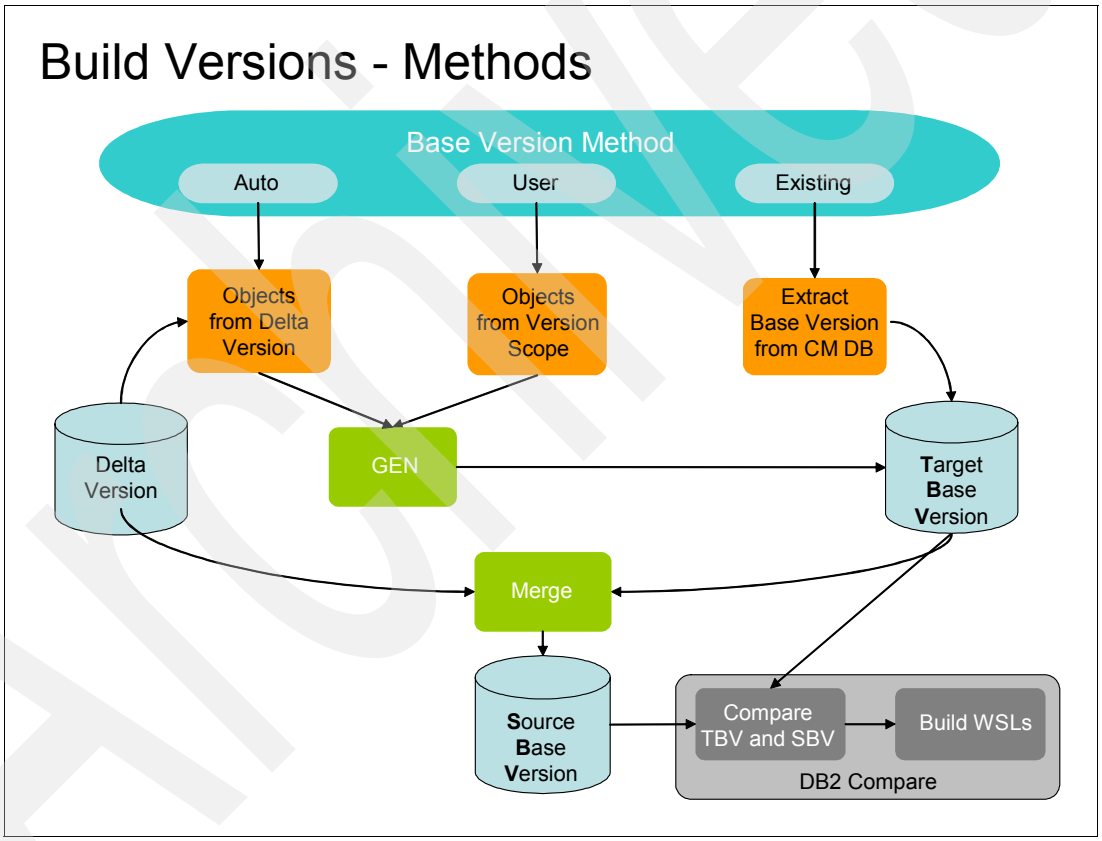

Figure 4-34 The version methods

If the change has prerequisites changes, the base version method option is forced according to the following criteria:

- AUTO is forced if neither of the conditions below are met.
- USER specified is forced if all the prerequisites have a status of ANALYZED and use the same version scope. The same version scope is used for the change that you are attempting to analyze.

- EXISTING base version is forced if all the prerequisites have a status of ANALYZED and use the same base version. The same base version is used for the change that you are attempting to analyze.

► PDS for WSL:

Once you have entered the data set name for your WSL, it remains inside your personal ISPF profile. So you are able to reuse the same data set when you analyze another change.

> **Note:** The data set for your WSL is especially important for a later run of your change. When you enter the ANALYZE process for the first time and the given WSL data set does not exist, Admin creates one for you. Make sure that this data set is still around when you want to run your change, as there is no way to tell the product to use existing SMS definitions. Refer to 4.1.2, “Data sets” on page 112, too.

► Validate WSL:

You might want to validate your WSL that is being created during the ANALYZE process. First, the validate function of ANALYZE can check the statements in your WSL but does not check all the statements in a cross-reference manner. You cannot validate a WSL being created during the ANALYZE process if your change has prerequisite changes.

Although validate and ANALYZE are supposed to create a WSL for your change which is able to run, there might be situations where an abend occurs while you are executing your change. For instance, if you drop a column from a table and there is a view referring to that table and column, the ANALYZE process, even included with validate, does not detect this dependence and you most likely receive an SQL code when running your change. Neither validate nor ANALYZE can check the status of an object today. Finally, although Validate and ANALYZE are good partners, you should maintain reasonable expectations.

► Run REBIND:

You might want to run DSN REBIND statements after you placed your DDL in DB2. If so, set REBIND to YES. The packages are taken from your existing DB2 environment where you run the ANALYZE job.

If everything works fine during ANALYZE, it finally builds a work statement list (WSL). This WSL is written in the data set you provide in the Analyze panel. Note that you cannot give a member name for your WSL. The member name is automatically generated. The first letter is always ‘C’ following by a 7 digit number. This number is your change ID given by Change Management during registration of the change. The example in Figure 4-35 shows the given naming convention.

```
DB2 Admin ----------- Work Statement List Library: 'ADMIN.CM.WSL' Row 38 of 55
Command ===>                                                   Scroll ===> CSR

Line commands:
 S - Show  R - Run in batch  D - Delete  C - Copy  A - Append  Q - Clone
 I - Interpret  V - Validate  E - Edit

Sel Name      Size Created    Changed          ID
    C000020*     * *          *                *
--- -------- ----- ---------- ---------------- --------
    C0000202     8 2007/03/14 2007/03/14 20:44 PAOLOR1
    C0000204    11 2007/03/15 2007/03/15 12:07 PAOLOR1
    C0000205    88 2007/03/15 2007/03/15 12:25 PAOLOR2
    C0000207    11 2007/03/15 2007/03/15 12:47 PAOLOR1
****************************** END OF DB2 DATA ******************************
```

Figure 4-35 Standard WSL member names are generated during ANALYZE

You can look at this work statement list with the command **WSL**. Be aware that modifying this work statement list might force a later RUN of your job to fail in the runtime analyze step. Therefore, we do not recommend modifying the created work statement list that has been generated during analyze. Change Management cannot track your changes if you run them outside the Change Management processes. Refer to “Runtime analyze” on page 147 to learn more about the effects when changing a WSL.

There are changes that cannot be analyzed because they are supposed to be executed fast. Fast changes are immediate and emergency changes. They are described in 4.11.5, “Emergency change” on page 180 and 4.11.6, “Immediate change” on page 184.

If a change has been analyzed and the WSL has been built and the object of interest is then deleted outside of Change Management, there is no automatic procedure available that tells Admin about the fact that this item has been deleted. So the status of this change remains in ANALYZED. However, if you try to run the change, runtime analyze is executed and the dropped object is then detected.

You can do a re-analyze of an analyzed change. Be aware that a change which has been analyzed before and is supposed to be analyzed again goes back into DEFINED status. If everything works fine during the re-analyze, the status is again ANALYZED. In terms of any problems, for instance missing objects, the Analyze process prints out an error message and keeps the status DEFINED for the change.

Be aware that the ANALYZE process creates temporary files. Make sure that they are around at least until the RUN process is finished. Your WSL executed in your RUN job might at least refer to the IFF data set. Because these temporary files are allocated in your ANALYZE job without a management class, they are just allocated on SYSDA volumes. We show an example of all these temporary files in Figure 4-36.

```
DSLIST - Data Sets Matching ADMIN.DB8A.C0000287                    Row 1 of 3
Command ===>                                                  Scroll ===> CSR

Command - Enter "/" to select action                 Message          Volume
-----------------------------------------------------------------------------
         ADMIN.DB8A.C0000287.AN162342.CHANGES                          SBOX79
         ADMIN.DB8A.C0000287.AN162342.IFF                              SBOX20
         ADMIN.DB8A.C0000287.AN162342.SHRVARS                          SBOXE1
***************************** End of Data Set list ****************************
```

Figure 4-36 Temporary files during ANALYZE process

The first qualifier is the qualifier that you can set for temporary data sets in the ANALYZE panel. The next three are generated by the ANALYZE process, you cannot make changes to them.

Here are some commands you might find useful. You can easily show the statements that belong to a change by typing **ST** as a line command. Depending on your change, you might see not just DDL, but DDL preceded by the keyword ADMIN. This is an Admin internal language; it is a meta language. It instructs Admin to arrange a more complex command stream in your WSL. The command **EA** lets you jump directly into the ANALYZE JCL.

Note: When you ANALZYE a change, you usually go directly to the ANALZYE batch job. However, if the DSNDB07 work file database is not available, you do not go to the batch job, and an error message is displayed instead indicating that DSNDB07 is not available. DB2 Admin requires 4 KB and 32 KB of work file space. Figure 4-37 shows an example of the error message that you would receive.

```
ADB1223E Unexpected sqlcode in : ADBCPRE:Insert into <dgtt>.ADBCHG
 DSNT408I SQLCODE = -904, ERROR:  UNSUCCESSFUL EXECUTION CAUSED BY AN
          UNAVAILABLE RESOURCE. REASON 00C90084, TYPE OF RESOURCE 00000230, AND
          RESOURCE NAME 4K
```

Figure 4-37 Error message provided if DSNDB07 is not available before ANALYZE

Important: The ANALYZE process marks your change in status as ANALYZED. In case of unexpected results or problems during ANALYZE, your change remains as DEFINED.

4.4.4 The RUN process

The run process in general looks as diagrammed in Figure 4-38.

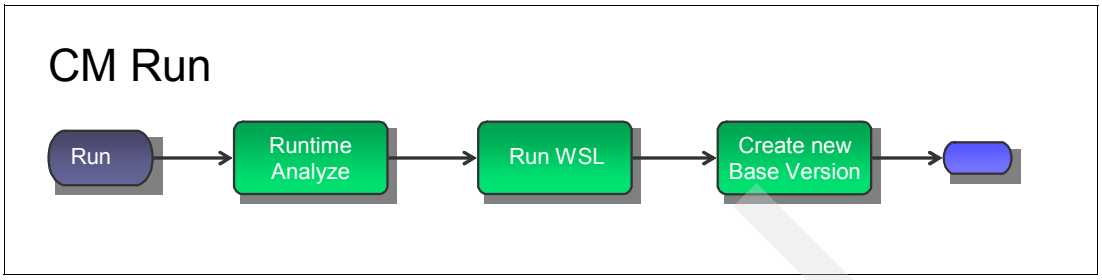

Figure 4-38 Run process

After a change has been analyzed, you can run this change easily by typing **RN** as a line command. When you run a change, a job is created that runs the work statement list (WSL) that was generated during ANALYZE, which you then submit. Do not run your WSL outside Change Management.

For a change that has been analyzed and is yet to be run, you can find information about the WSL that was generated by entering the **I** line command before the change. See Figure 4-39. The name of data set for the WSL is displayed. As the run job has not been created yet, the field for the name of the data set for the run batch job is blank. The run job is created after you enter **RN** as a line command.

If you enter the **RN** command and you then go back and enter **I** in front of your change again, you can see that the field, RUN job DSN, is filled out now. The membername for your run job always contains your change ID preceded by **E**.

```
DB2 Admin ------------- CM - Interpretation of a Change ----------------- 20:47
Command ===>

Details for change : PAOLOR1.D070315-CR#63-NOR

Change ID  . . . : 231            Status . . . : ANALYZED
Created by . . . : PAOLOR1        Created  . . : 2007-03-15-17.58.54.617528
Last altered by  : PAOLOR1        Last altered : 2007-03-15-18.56.10.747850
Change type  . . : CHANGE
WSL DSN  . . . . : 'ADMIN.CM.WSL(C0000231)'
Run job DSN  . . :
Analyze job DSN  : 'ADMIN.CM.CNTL(C0000231)'
Comment  . . . . :



                   Owner         Name
Change . . . . . . : PAOLOR1     : D070315-CR#63-NOR
Mask . . . . . . . : ?           : ?
Ignore . . . . . . : ?           : ?
Version scope. . . : ?           : ?
Base version . . . : ?           : ?
Source version . . : ?           : ?
Target version . . : ?           : ?
Delta version. . . : PAOLOR1     : D070315-CR#63-NOR
```

Figure 4-39 Information about a change before entering the RN command

Notice that once the run job has been submitted, the change status changed from ANALYZED to RUNNING. You can see this by invoking the Changes Overview panel while the job is still active. If everything works fine with your change and the job ends normally, the status changes from RUNNING to COMPLETE.

```
DB2 Admin ---------------- CM - Run a Change --------------------------------
Command ===>


Change . . . : PAOLOR1.D070315-CR#63-NOR


Data set information:
  PDS for batch jobs . . . . . 'ADMIN.CM.CNTL'
  Prefix for data sets . . . . ADMIN

Create a new base version  . . NO   (Yes/No)

  Scope Information:
    Owner  . . . . . . . . . .          >           (? to look up)
    Name . . . . . . . . . . .                    >  (? to look up)

  Version Information:
    Owner  . . . . . . . . . .          >           (? to look up)
    Name . . . . . . . . . . .                    >  (? to look up)

Runtime analyze:
  Change reporting options . . NO   (Yes/No)
```

Figure 4-40 The RUN panel

Runtime analyze

When you run a a change, a mandatory task is performed. This task is called Runtime WSL. The Run job creates a second WSL of your activities and compares this WSL with the existing one.

Runtime analyze is a mandatory feature to make sure that nothing has changed from the moment the change has been analyzed. It is an internal quality assurance for your change. You cannot jump over it, it is always executed when you want to run a change.

Make sure that you do not modify your WSL before you run your job. Otherwise you would most likely get the error shown in Figure 4-41.

Archived

```
 Display  Filter  View  Print  Options  Help
 ------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY ADBXXXXX JOB21798  DSID   126 LINE 0       COLUMNS 02- 81
 COMMAND INPUT ===>                                            SCROLL ===> CSR
********************************* TOP OF DATA *********************************
db2aloc  =
db2serv  =DB8A
db2auth  =PAOLOR1
db2rel   =815
descript =
adbtest  =YES
adbtstpn =A72
srcwslst =C0000207
srcwslib =ADMIN.CM.WSL
voption  =
ADBC099E: WSL mismatch. The WSLs did not compare equally.
ISPD117
The initially invoked CLIST ended with a return code = 8
 PAOLOR1.ADBXXXXX.JOB21798.D0000124.? was preallocated (no free was done).
READY
END
******************************** BOTTOM OF DATA ******************************
```

Figure 4-41 The output shows that runtime analysis detects a mismatch

The change status remains in status ANALYZED. There is no change to the status because your change does not start to execute if runtime analysis fails.

Be aware that this is not a tool defect. You come across this situation when your existing WSL and the generated runtime WSL are not in sync. In this case, you have to re-analyze your change again, then do the **RN** command again and run your change.

Restarting a RUN job

Situations can occur where your WSL can end with a RC greater than 4. In these case, your change remains in the status RUNNING. It is NOT set to another state. The state RUNNING remains because the status changes only as a last step in your run job if everything works fine.

In case of an abended RUN job, you have to set up the RUN job again. Analyze the joblog and make sure that you eliminate the reason for the abended change job. There is a hidden “restart” command available. Just type **ER** as a line command (Edit Run job). You can then manually edit the job and submit it again. You have to restart the job at the step which contains the WSL. Notice that it is not always the same stepname. In our example in Figure 4-42, we had to restart the job at step DB2B7.

```
DB2 Admin ------------------ CM - Changes ------------------------- Row 9 of 9
Command ===>                                                   Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- ---------------------
ER       243 PAOLOR1  D070316-CR#74-CMP  CHANGE  RUNNING  OBJECT COMPARE
****************************** END OF DB2 DATA ******************************
```

Figure 4-42 Restarting a change with ER

After typing **ER** , you have to modify the RESTART step as described in Figure 4-43.

```
 EDIT       ADMIN.CM.CNTL(E0000243) - 01.02                Columns 00001 00072
Command ===>                                                   Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 //ADBXXXR2 JOB (),'ADM TOOL',REGION=0M,
000002 //        RESTART=DB2B7, <== FOR RESTART REMOVE * AND ENTER STEP NAME
000003 //         MSGCLASS=S,NOTIFY=&SYSUID,
000004 //         CLASS=A
000005 //*
000006 /*JOBPARM S=SC63
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -   345 Line(s) not Displayed
000352 //*********************************************************************
000353 //*
000354 //* DB2 BATCH MONITOR
000355 //*
000356 //* DB2 ADMIN GENERATED BATCH JOB.
000357 //*
000358 //*************************************************************ADB2WL4**
000359 //DB2B7 EXEC PGM=IKJEFT01,DYNAMNBR=100
000360 //STEPLIB  DD DISP=SHR,DSN=DB8A8.SDSNEXIT
000361 //         DD DISP=SHR,DSN=DB8A8.SDSNLOAD
000362 //SYSEXEC  DD DISP=SHR,DSN=GOC.V7R2M0.SGOCEXEC
000363 //         DD DISP=SHR,DSN=ADB.V7R2M0.SADBEXEC
000364 //MSGLIB   DD DISP=SHR,DSN=GOC.V7R2M0.SGOCLLIB
000365 //         DD DISP=SHR,DSN=ADB.V7R2M0.SADBLLIB
000366 //SYSTSPRT DD SYSOUT=*
000367 //SYSPRINT DD SYSOUT=*
000368 //UTPRINT  DD SYSOUT=*
000369 //RNPRIN01 DD SYSOUT=*
000370 //ADBDIAG  DD SYSOUT=*
000371 //SYSTSIN  DD *
```

Figure 4-43 Modifying the RESTART step of the job

Base Version during RUN

You have the opportunity to generate a base version automatically after your change has been executed. This helps to ensure that your recently applied changes are captured automatically after a successful RUN. In this case, you have to specify a new base version and the existing version scope. PK38750 should be installed prior using this feature. If you plan to deploy your changes later in other DB2 environments using the PROMOTE process, we recommend that you generate a new base version after the changes have been applied. See Figure 4-44.

```
DB2 Admin ---------------- CM - Run a Change ----------------------------------
Command ===>


Change . . . : PAOLOR1.D070315-CR#63-NOR


Data set information:
  PDS for batch jobs . . . . . 'ADMIN.CM.CNTL'
  Prefix for data sets . . . . ADMIN

Create a new base version  . . YES  (Yes/No)

  Scope Information:
    Owner  . . . . . . . . . . .          >          (? to look up)
    Name . . . . . . . . . . . .                   >  (? to look up)

  Version Information:
    Owner  . . . . . . . . . . .          >          (? to look up)
    Name . . . . . . . . . . . .                   >  (? to look up)

Runtime analyze:
  Change reporting options . . NO   (Yes/No)
```

Figure 4-44 Taking a new base version during RUN

If you decide to create a new base version during run, make sure that you enter the version scope and a new name for a new base version. Usually you want to keep your base versions. If you try to use an already existing base version, you are asked if you want to replace the existing base version.

When you create a new base version during the RUN process, the base versions are not protected by default. There is an option to protect your base versions from being deleted accidentally. When displaying your versions from the Manage Versions Menu, you can use the line command **PT** to switch from protected mode to unprotected mode for a single version. You can use the line command **I** to see if a version is protected or not. See Figure 4-45.

Important: The RUN process marks your change either in status RUNNING or COMPLETE.

```
DB2 Admin ------------- CM - Interpretation of a Version ---------------- 16:12
Command ===>

Details for version : 'PAOLOR1'.'DGDB0290 BASE 2'

Version ID  . . : 204
Version owner . : PAOLOR1          Version name : DGDB0290 BASE 2
Created by  . . : PAOLOR1          Created  . . : 2007-03-15-21.20.00.000000
Last altered by : PAOLOR1          Last altered : 2007-03-15-21.21.13.602121
Protected . . . : Yes              Type . . . . : Base
Location  . . . :                  Scope ID . . : 22
Mixed . . . . . : N                Server . . . : DB8A
Date format . . : ISO              Time format  : ISO
DB2V8NF . . . . :                  Encoding Scheme : E

GEN SSID  . . . :                  GEN REL  . . . . . : 810
DECP REL  . . . : 810              DECIMAL POINT  . . : .
DECP SSID . . . : DB8A             EBCDIC CCSID . . . : 37
EBCDICM CCSID . : -2               EBCDICG CCSID  . . : -2
ASCIIS CCSID  . : 437              ASCIIM CCSID . . . : -2
ASCIIG CCSID  . : -2               UNICODES CCSID . . : 367
UNICODEM CCSID. : 1208             UNICODEG CCSID . . : 1200

Comment . . . . :
```

Figure 4-45 Protection status of a version

4.4.5 The PROMOTE process

In this section we discuss the PROMOTE process in depth. The PROMOTE process itself always follows the workflow as described in Figure 4-46.

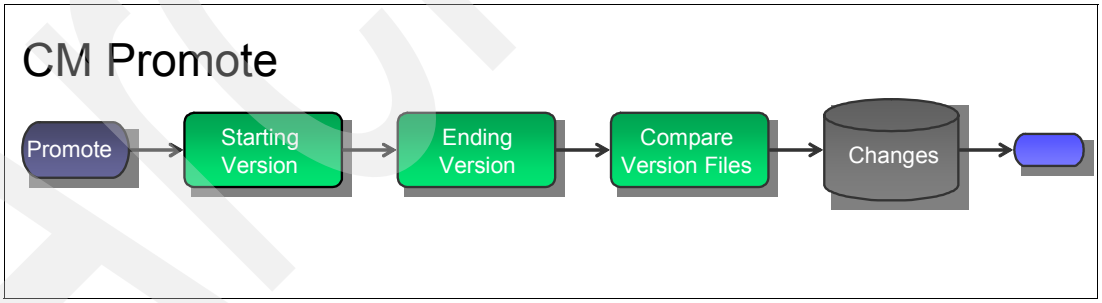

Figure 4-46 The PROMOTE workflow

Many customer environments consist of different DB2 subsystems each representing different levels of application development. For instance, you want to execute a change which has already been done in your development subsystem now in your quality assurance subsystem.

Therefore Change Management offers the ability to promote a change from a source system to as many target systems as you have to. PROMOTE is the process which has to be done for a delta change.

PROMOTE compares two base versions. Base versions are just internal representations of DDL. The starting base version is the version before the change has been applied. Vice versa, an ending version is the version after the change has been applied. PROMOTE summarizes the differences between the base versions and stores these differences in a data set called a *delta changes data set*, which contains more than just DDL. Depending on your change, it also contains Admin's meta language.

If you plan to deploy your change to multiple targets, you have to import your change in every target environment.

To invoke the PROMOTE process, from the panel shown in Figure 4-47, you would select option **3** - Create delta for target.

```
DB2 Admin ---------------------  CM - Manage Changes -------------------- 17:49
Option ===> 3


   1 - Display changes                                  DB2 System: DB8A
   2 - Create a change                                  DB2 SQL ID: PAOLOR1
   3 - Create delta for target
   4 - Import changes









Enter display selection criteria (Using a LIKE operator, criteria saved):
Name . . . . . .                    >  Created by . . PAOLOR1% >
Owner  . . . . . .          >          Altered by . .          >
Type . . . . . .                       Status . . . .
Created within . . 1 DAYS              Change ID  . .
Altered within . .
```

Figure 4-47 Invoking the PROMOTE process

Archived

You then do a compare between a starting base version and an ending base version. Therefore, if you plan to use the PROMOTE process, you should take a base version before you register the change, and then another base version after the change has been implemented successfully. See Figure 4-48.

```
DB2 Admin ----------------------- CM - Promote -------------------------- 18:00
Command ===>

Starting Version (Old):
  Owner . . . . . . PAOLOR1  >            (? to look up)
  Name  . . . . . . D070319-CR#80-BS1  >  (? to look up)

 or enter a data set name that contains a Start Version:

  Data set name . .

Ending Version (New):
  Owner . . . . . . PAOLOR1  >            (? to look up)
  Name  . . . . . . D070319-CR#80-BS2  >  (? to look up)

Output data set names:
  Promote JOB JCL . 'ADMIN.CM.CNTL.PROMOTE'
  Delta change  . . 'ADMIN.CM.DELTA1'
```

Figure 4-48 Where to enter your BASE versions while doing the PROMOTE process

Note: The same panel as described in Figure 4-48 could be invoked if you type **PR** (PROMOTE) as a line command in front of a base version. The base versions can be listed with the command **VE** in front of a version scope. You can refer to Chapter 7.1, “Rolling out using the PROMOTE process” on page 312 to get more information about a real life scenario for PROMOTE.

The PROMOTE process provides you the differences between your two base versions. The “delta” is stored in a delta changes data set, in our example the data set name is ADMIN.CM.DELTA1.

Archived

Creating a delta changes data set with the PROMOTE process is a change as well. You are asked to enter the change name, you do not have to enter a delta version for this change. See Figure 4-49.

```
DB2 Admin -------------------- CM - Register Options -------------------- 18:30
Option ===>

Commands: CONTINUE                                        DB2 System: DB8A
                                                          DB2 SQL ID: PAOLOR1
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR1  >  (Optional, default is PAOLOR1)
Name  . . . . . . . . D070316-CR#86-NOR                                  >
Comment . . . . . . .                                                    >
Change Type . . . . : COMPARE  (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                      Owner         Name
Ignore  . . . . . . .         >                          >  (? to look up)
Mask  . . . . . . . .         >                          >  (? to look up)
Delta Version . . . .         >                          >  (? to look up)
```

Figure 4-49 A change for the PROMOTE process

You can look up your change which is in status DEFINED, as shown in Figure 4-50.

```
DB2 Admin ----------------- CM - Changes ------------------------ Row 8 of 8
Command ===>                                                 Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name               Type    Status   Comment
           * *        *                  *       *        *
--- -------- -------- ----------------- ------- -------- --------------------
         259 PAOLOR1  D070316-CR#86-NOR  COMPARE DEFINED
****************************** END OF DB2 DATA ******************************
```

Figure 4-50 When comparing two base versions, the corresponding change is DEFINED

Finally, JCL is generated to create the delta change data set.

Notes:

- If you leave the JCL and jump into the change summary, your change is still in the status DEFINED. There is just the command **ER** available, which lets you invoke the PROMOTE job again to create the delta change data set.
- You have to submit the generated JCL in order to bring your change in a COMPLETE status. Do not delete the delta change data set, as it will be allocated when you are still in the dialog and not in the provided JCL.
- When using the PROMOTE process, it is not possible to ANALYZE the related change.

The generated JCL fills your delta change data set with DDL. As an example, let us look into the currently generated delta changes data set shown in Figure 4-51.

```
VIEW       ADMIN.CM.DELTA1                                  Columns 00001 00072
Command ===>                                                   Scroll ===> CSR
000001 DROP      ADMIN DROP TABLE DSN8810.EMP ;
000002 CREATE    CREATE TABLE DSN8810.EMP
000003 CREATE       (EMPNO                CHAR(6) FOR SBCS DATA NOT NULL,
000004 CREATE        FIRSTNME             VARCHAR(12) FOR SBCS DATA NOT NULL,
000005 CREATE        MIDINIT              CHAR(1) FOR SBCS DATA NOT NULL,
000006 CREATE        LASTNAME             VARCHAR(15) FOR SBCS DATA NOT NULL,
000007 CREATE        WORKDEPT             CHAR(3) FOR SBCS DATA WITH DEFAULT NU
000008 CREATE        PHONENO              CHAR(4) FOR SBCS DATA WITH DEFAULT NU
000009 CREATE        HIREDATE             DATE WITH DEFAULT NULL,
000010 CREATE        JOB                  CHAR(8) FOR SBCS DATA WITH DEFAULT NU
000011 CREATE        EDLEVEL              SMALLINT WITH DEFAULT NULL,
000012 CREATE        SEX                  CHAR(3) FOR SBCS DATA WITH DEFAULT NU
000013 CREATE        BIRTHDATE            DATE WITH DEFAULT NULL,
000014 CREATE        SALARY               DECIMAL(9, 2) WITH DEFAULT NULL,
000015 CREATE        BONUS                DECIMAL(9, 2) WITH DEFAULT NULL,
000016 CREATE        COMM                 DECIMAL(9, 2) WITH DEFAULT NULL,
000017 CREATE        CONSTRAINT EMPNO
000018 CREATE        PRIMARY KEY (EMPNO),
000019 CREATE        CONSTRAINT NUMBER
000020 CREATE          CHECK (PHONENO >= '0000' AND PHONENO <= '9999'),
000021 CREATE        CONSTRAINT PERSON CHECK (SEX = 'M' OR SEX = 'F'))
000022 CREATE      IN DSN8D81A.DSN8S81E
000023 CREATE      PARTITION BY (EMPNO ASC)
000024 CREATE       (PART 1 VALUES('099999'),
000025 CREATE        PART 2 VALUES('199999'),
000026 CREATE        PART 3 VALUES('299999'),
000027 CREATE        PART 4 VALUES('499999'),
000028 CREATE        PART 5 VALUES('999999'))
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 10 Line(s) not Displayed
****** *************************** Bottom of Data ****************************
```

Figure 4-51 Delta change generated DDL

Important: The PROMOTE process marks your change either in status DEFINED or COMPLETE.

4.4.6 The IMPORT process

In this section we discuss the IMPORT process in depth. The IIMPORT process itself always follows the workflow as described in Figure 4-52.

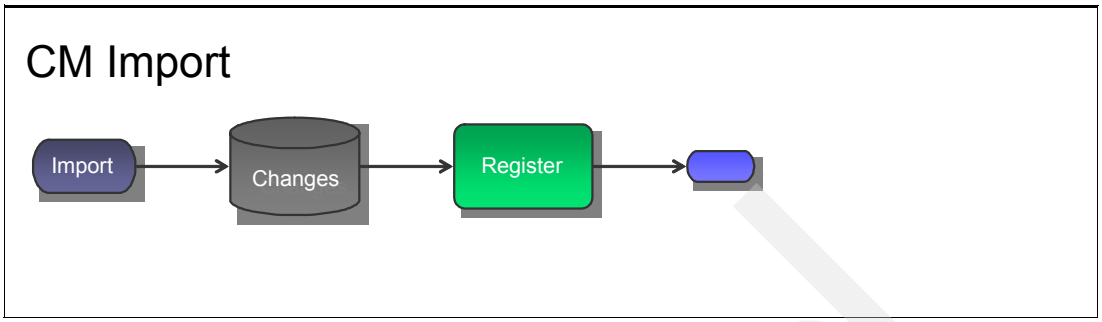

Figure 4-52 The IMPORT workflow

The IMPORT process provides you the ability to register changes from a data set containing DDL. You can import:

- DDL that is provided by the GEN command of Admin or any other DDL generation tool of your choice
- A delta changes data set that has been produced as a result of:
  - The PROMOTE process
  - Compare

If you plan to import your delta change data set (or your DDL data set) to multiple targets, you have to import your DDL or delta change data set in every target environment where you want to run the imported change.

While importing a change, you can use masking to adapt the DDL to your naming standards.

Restriction: There is currently no overwrite available during the IMPORT process, as it is possible to do this in the WSL cloning process.

According to the IMPORT process itself, it does not make any difference if you provide native DDL, an output data set from Compare, or an output data set from the PROMOTE process.

To start the import, use option **1** - Manage Changes and then **4** - Import changes. You are required to enter the data set name. Put it in quotes to avoid conflicts. See Figure 4-53.

```
DB2 Admin ------------------- CM - Import Changes ----------------------- 20:26
Command ===>


Input data set information:                              DB2 System: DB8A
 Data set name 'ADMIN.DB8A.C0000246.AN195814.CHANGES'

Prereq resolution mode  . B         (TSO/Batch)
Execution mode  . . . . . B         (TSO/Batch)
```

Figure 4-53 The IMPORT screen

Change Management makes sure that prerequisites are being detected during the IMPORT process. We recommend that you start the IMPORT in batch mode, then you must do the prereq resolution mode in batch as well.

You are prompted to tell Change Management what kind of action do you want to perform if the change you are going to import has pending prerequisite changes (see Figure 4-54).

If so, you can then do any of the following operations:

- CANCEL:

  Cancel the change being imported.
- PREREQ:

  Make the pending changes for the objects prerequisite changes for the imported change.
- SUPERSEDE:

  Make the imported change a prerequisite change for the pending changes.

```
DB2 Admin -- DB8A CM Import a Change - Action for Pending Changes - 20:35

Specify an action to use if the imported change has pending prerequisite




Select a choice
  1.  Cancel
  2.  Prereq
  3.  Supersede
```

Figure 4-54 Action for pending changes

Using Masking during the IMPORT process

You can adapt your naming standards while doing the IMPORT process. In the upcoming panel, which is the register panel of your change, use a masking of your choice to adapt naming standards. See Figure 4-55.

```
DB2 Admin -------------------- CM - Register Options -------------------- 20:51
Option ===>

Commands: CONTINUE                                      DB2 System: DB8A
                                                        DB2 SQL ID: PAOLOR1
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR1  >  (Optional, default is PAOLOR1)
Name  . . . . . . . . D070316-CR#78-IMP                                    >
Comment . . . . . . . IMPORT                                               >
Change Type . . . . : PROMOTE  (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                      Owner         Name
Ignore  . . . . . . .         >                          >  (? to look up)
Mask  . . . . . . . . PAOLOR1  >    PROT.DBAF.MASK.04    >  (? to look up)
Delta Version . . . .         >                          >  (? to look up)
```

Figure 4-55 Using Masking while doing the IMPORT process

Point-and-shoot to CONTINUE and your IMPORT job is created. Then submit.

Important: The IMPORT process sets your change to status DEFINED.

4.4.7 The RECOVER process

In this section we discuss the RECOVER process in some detail. The RECOVER process itself always follows the workflow as described in Figure 4-56.

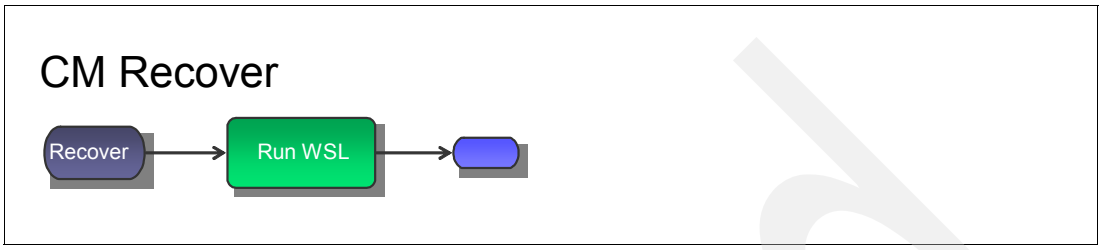

Figure 4-56 CM Recover Process

You can recover changes that have been made through the Change Management. When you analyze a change, you have the option of having a recover change created. DB2 Admin generates a recover change, assigns a change ID to the recover change, and when the analyze job finishes, it puts the recover change (and the change itself) in ANALYZED status.

To recover a change, you issue the **RC** line command for the original change. You do *not* issue the **RN** line command for the recover change.

Note: Fast changes cannot be recovered. Because fast changes are run immediately upon registration, you cannot analyze or run them manually.

DB2 Admin determines the requirements for backing out a completed change:

- Changes must be in COMPLETE status.

  A recover change must exist. When you analyze a change, you can optionally specify that a recover change be created automatically. When you first choose to create a recover change, you are prompted to register the recover change. The recover change is automatically updated if the original change is reanalyzed.
- A WSL that was generated for the recover change during analyze process must also be available.
- Whether the change has a recover change is determined.
- Pending changes to the objects being affected by the change that must be reanalyzed after the change is recovered are also considered.

Recover of other completed changes must be recovered first and must be performed in a specific order.

When generating a recover change, you have to decide the scope of the recovery action by choosing between the original data and the existing data:

- **Original data:**

  Original data is the data that exists just before the original change is run.

  This option applies only to tables that were dropped for the original change, regardless of whether they were re-created in the change.

  Original data can be recovered only for objects that are dropped as part of the original change; referential integrity is not considered. Original data cannot be recovered for changes that are made using an SQL ALTER or RENAME statement.

- **Existing data:**

  Existing data is the data that exists in the table just before the original change is recovered.

  If a table is dropped without being re-created in the original change, no data is loaded after the table is created in the recover change.

Table 4-4 gives an overview of the various possibilities existing when changing columns.

Table 4-4 Change action for table columns and resulting data recover generations

Change action for table	Original data generated in WSL	Existing data generated in WSL	Recover action for tables
CREATE TABLE	No	No	DROP TABLE
DROP TABLE	Yes	No	CREATE TABLE
ALTER TABLE ADD COLUMN	No	Yes	DROP/CREATE Table
ALTER TABLE increase column length	No	Yes	DROP/CREATE TABLE
ADD Column (not at the end)	Yes	Yes	DROP/CREATE
DROP column (at the end)	No	No	ALTER TABLE ADD COLUMN
DROP column (not at the end)	Yes	Yes	DROP/CREATE TABLE
RENAME column	Yes	Yes	DROP/CREATE TABLE
Decrease column length	No	No	ALTER TABLE increase column length
Change data type	Yes	Yes	DROP/CREATE TABLE

As you can see, there is not always a recovery possible for original data. The same applies for existing data. In the situation where multiple tables are processed in a WSL, then you can end up with mixed data — original and existing. This result is probably not desirable, as it can lead to data integrity issues.

Attention:

- If your change action does not result in a drop of the table, then there is no original data available. An ALTER TABLE ADD COLUMN at the end of the table generates a recover WSL to existing data — although original data was chosen. There is no warning to make you aware of this fact!
- If your change contains a mix of tables (tables that get dropped and other tables that just get SQL ALTERs) and you specify to recover with ORIGINAL data, then you have a mix of ORIGINAL data and EXISTING data!

During the analyze process, we can specify if we want to generate a recover change.
We entered **YES** on the Generate Analyze Job panel as shown in Figure 4-57 (this option is forced to YES if the change has been previously analyzed and has a recover change).

```
DB2 Admin ----------------- Generate Analyze Job  ----------------------- 13:44
Command ===>

 Specify the following for Analyze:

 Base version method . . . . . A        (Auto, User, or Existing)

 Change reporting options  . . NO       (Yes/No)

 Required data set information:
   PDS for WSL . . . . . . . ADMIN.CM.WSL
   PDS for batch jobs  . . . ADMIN.CM.JCL
   Prefix for data sets  . . PAOLOR3

 Options:
   Run SQLID . . . . . . . .          (Blank, a SQLID, or <NONE>)
   Validate WSL  . . . . . . NO       (Yes/No)
   Use utility options . . . NO       (Yes/No)
   Generate templates  . . . NO       (Yes/No)
   Build JCL to run WSL  . . YES      (Yes/No)
   Generate a recover change YES      (Yes/No)
     Data to recover . . . . E        (Original or Existing)


   Optional jobs after Reload or Alter:
     Run CHECK DATA  . . . . NO       (Yes/No)
     Take an image copy  . . N        (after: Reload/Alter/Both/None)
     Run REORG . . . . . . . N        (Mandatory, All relevant, None)
     Run RUNSTATS  . . . . . N        (after: Reload/Alter/Both/None)
     Run REBIND  . . . . . . NO       (Yes/No)

 BP - Change batch job parameters
 TU - Specify TEMPLATE usage
 UO - Change utility options
```

Figure 4-57 Generate Analyze Job panel with recover option

The job to perform the analysis is generated afterwards. After submitting it, a recover WSL was created and the recover change was registered in the CM database. See Figure 4-58.

```
DB2 Admin ------------------ CM - Changes ------------------- Row 134 from 134
Command ===>                                                  Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * PAOLOR3* *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
         163 PAOLOR3  REC CHNG FOR C0149 RECOVER ANALYZED RECOVER CHANGE FOR CH
****************************** END OF DB2 DATA ******************************
```

Figure 4-58 Generate Analyze Job panel with recover option

We execute the original change and the change status is set to COMPLETE. See Figure 4-59.

```
DB2 Admin ------------------ CM - Changes ------------------- Row 121 from 136
Command ===>                                                   Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * PAOLOR3* *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
RC       149 PAOLOR3  REN COL LASTNAME   CHANGE  COMPLETE REN COL LASTNAME IN T
         163 PAOLOR3  REC CHNG FOR C0149 RECOVER ANALYZED RECOVER CHANGE FOR CH
****************************** END OF DB2 DATA ******************************
```

*Figure 4-59 CM Changes panel showing status for change type **CHANGE** and **RECOVER***

After successful change recover processing, the change is now in COMPLETE status. The status for the original change is set to DEFINED (see also Figure 4-60).

```
DB2 Admin ------------------ CM - Changes ------------------- Row 130 from 147
Command ===>                                                   Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
        *    PAOLOR3* *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
         149 PAOLOR3  REN COL LASTNAME   CHANGE  DEFINED  REB COL LASTNAME IN T
         163 PAOLOR3  REC CHNG FOR C0149 RECOVER COMPLETE RECOVER CHANGE FOR CH
****************************** END OF DB2 DATA ******************************
```

Figure 4-60 CM Changes panel showing status after recover completed

You can find scenarios of the RECOVER process in Chapter 8, “Recovery of changes” on page 329.

We also want to mention recovering a change that has not yet completed. For this situation (recovering an incomplete change), we still require procedures in place to revert such a change. The product does not match this situation yet by an existing process. However, a combination of manual interventions is required to facilitate the recovery of an uncompleted change. But it really depends on the nature of the failure how we can recover from such a problem. A sample scenario for this is described in 8.3, “Recovering an uncompleted change” on page 354.

4.5 Change status

The status of a change is updated when actions are taken by the user of the change. Table 4-5 describes the possible values for the status.

Table 4-5 Status of changes

Status	Description
INITIAL	The change has been created, but its registration in the Change Management database is incomplete. You can try to get the change registered by issuing the restart line command (RST) on the Changes panel (ADB2C11). However, DB2 Admin cannot detect and process any prerequisite changes that might exist. You have to identify any prerequisite changes yourself and reanalyze any change in ANALYZED status to ensure its validity.
DEFINED	The change has been created and registered in the Change Management database. The change is ready to be analyzed.
ANALYZED	The change has been analyzed by verifying how the change modifies existing objects, both in DB2 and in any of the prerequisite changes, and a WSL has been generated. The change is ready to be run.
RUNNING	The change is currently being run. Possible outcomes would be: Complete - it ran successfully. Running - which does not change to COMPLETE status Indicates that the change failed at some point. Review job output to see if you can determine the cause of the failure. You can use 'Edit Run (**ER**) command to make changes. Make the necessary corrections, and restart the job at the appropriate step. If the job failed anywhere before the ADBTEP2 step, restart the job at the beginning. If the job failed at the ADBTEP2 step, restart at that step.
COMPLETE	The change has been run successfully, regardless if it was a normal or a fast change.
CANCELED	The change has been canceled. Changes can be left in CANCELED status since they do not affect other changes.
FAILED	The change is a fast change that was run immediately but did not complete successfully. Normally this is caused by a user error. Use the **I** command (Interpretation of a Change) to see details of the fast change. Changes can be left in FAILED status since they do not affect other changes.

Archived

4.6 Change types

DB2 Admin categorizes changes into several types and assigns a status to each change as it moves through the Change Management process. The type is assigned when a change is registered. Table 4-6 describes the types of changes.

Table 4-6 Type of changes

Type of change	Description	Processes this change can take
NORMAL	A change that is defined through the usual change functions in DB2 Admin and DB2 Object Comparison Tool, such as ALC, AL RDEF, and Compare.	Register, Analyze, Run, Recover
FAST	A change that is run immediately. If pending changes exist for the object or related objects that are affected by the fast change, the fast change is called an **emergency** change, and it supersedes the pending changes. Once this fast change takes place, the pending changes are placed in DEFINED status. If no pending changes exist, the fast change is called an i**mmediate** change. Fast changes cannot be analyzed nor run manually. You also cannot modify fast changes, recover them, or promote them to other systems	Immediate, Emergency
COMPARE	A change that is generated by comparing two items such as two base versions, two DDL files, two catalog objects, or a DDL file and a catalog object. It gets registered as a COMPARE change and only shows COMPLETE. It is only used as for bookkeeping on the source system. The delta file can be used on another system in the import process and becomes a PROMOTE change on the other system.	Created on the source. Register (as COMPARE) with status Complete. Delta file can be used on another system as import file to be Registered as a Promote change
PROMOTE	A change that is generated by importing statements from a data set or from the delta from COMPARE changes file.	Created on the target. Obtained from imported file or Compare delta file. Only registered as a PROMOTE change. Analyze, Run, Recover
RECOVER	A change that was automatically generated to back out another change. When you analyze a change, you have the option of having a recover change created. DB2 Admin generates a recover change, assigns a change ID to the recover change, creates a delta version for the recover change, and puts the recover change in ANALYZED status. To recover a change, you issue the RC line command for the original change. You do not issue the RN line command for the recover change.	

Archived

4.7 Canceling a change

You have the opportunity to cancel a change, regardless of its status, except for a COMPLETE change. A CANCELED change still remains in Change Management.

4.8 Restarting

If the registration fails, change header and change statements are already populated in the CM tables ADBCHG and ADBCHGS. The change status is INITIAL. To issue the restart command, the change must be in INITIAL status. Whenever a change is edited, it goes back to status INITIAL.

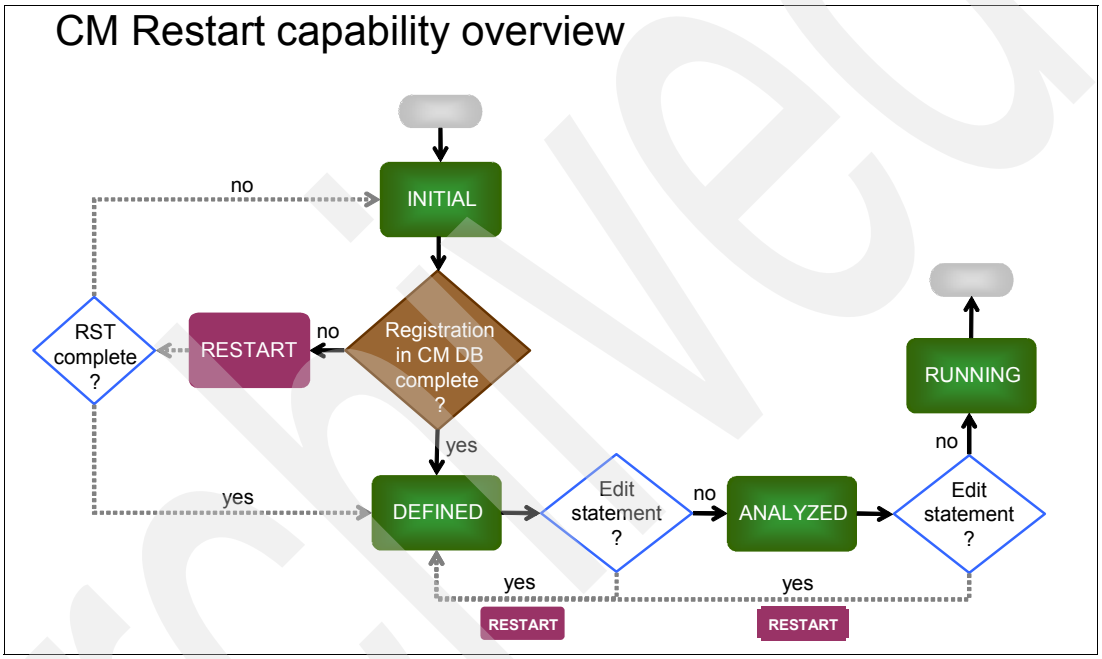

Figure 4-61 CM Restart capability overview

If the registration was not successful due to syntactic or semantic validation errors, you have to edit the change statements. In this case, the change is restarted automatically when leaving the panel as shown in Figure 4-63.

You must use caution while “Restarting a Change” and “Editing Change Statements”. If the change is INITIAL status and you issue the restart line command to attempt to complete it, DB2 Admin cannot detect and process any prerequisite changes that might exist. They are not rebuilt by DB2 Admin. Therefore you have to identify any prerequisite changes yourself and re-analyze any change in ANALYZED status to ensure its validity.

There are two ways that lead to RESTART mode:

- Using the **RST** line command
- Editing change statements using the **E** line command

In the Selection column on the CM - Changes panel, as shown in Figure 4-62, we entered the **RST** line command in front of the change.

```
DB2 Admin ------------------ CM - Changes --------------------- Row 54 from 56
Command ===>                                                  Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
RST       60 PAOLOR3  ALTER_COLUMN_DEPT  CHANGE  INITIAL  INCREASE COLUMN LENGT
****************************** END OF DB2 DATA ******************************
```

Figure 4-62 Using restart capability on the Changes panel

Change statements can be edited using the **E** line command on the panel, CM - Change Statements, as shown in Figure 4-63.

```
DB2 Admin ------------- CM - Change Statements ------------------ Row 1 from 1
Command ===>                                                  Scroll ===> CSR

Change statements for change "PAOLOR3"."ALTER_COLUMN_DEPT"
Line commands:
 E - Edit  D - Delete  I - Insert  S - Show

Sel    Sequence O  Qual     Name                Statement
              * *  *        *                   *
--- ----------- -- -------- ------------------ ------------------------------>

E             1 TB DSG24D03 GLWTDPT             ADMIN ALTER TABLE "DSG24D03"."G
****************************** END OF DB2 DATA ******************************
```

Figure 4-63 Modify change statement on the Change Statements panel

After inserting or modifying a statement, the object type (O), qualifier (Qual), and name (Name) columns are blank (see Figure 4-64). These values are automatically populated when the change is restarted. The change is automatically restarted upon exiting from the panel CM - Change Statements.

```
DB2 Admin ------------- CM - Change Statements ------------------ Row 1 from 1
Command ===>                                                  Scroll ===> CSR

Change statements for change "PAOLOR3"."ALTER_COLUMN_DEPT"
Line commands:
 E - Edit  D - Delete  I - Insert  S - Show

Sel    Sequence O  Qual     Name                Statement
              * *  *        *                   *
--- ----------- -- -------- ------------------ ------------------------------>

              1                                 ADMIN ALTER TABLE "DSG24D03"."G
****************************** END OF DB2 DATA ******************************
```

Figure 4-64 Change Statements panel after editing the statement

4.9 Using the masking feature

Masking is an important feature if you want to adapt the naming standards of your DDL inside a change. You can use the feature masking while you try to import normal DDL or perhaps a delta change data set. In the same way, masking can be used when you use Compare.

With Admin and Compare 7.2, the ability to store your masking items inside DB2 tables has been introduced. The affected table of your Change Management database is:

`:cmown.ADBCMASK`

This table contains the different masking names. A masking name consists of OWNER and NAME. Every entry gets a specific number inside this table:

`:cmown.ADBCMASKS`

According to the specific number from :cmown.ADBCMASK, there is a sequence entry in :cmown.ADBCMASKS that represents one masking change (comparable to one row in an existing masking data set).

As mentioned earlier in the book, mask files from 7.1 can be converted into CM mask files by creating a new mask from panel ADB2C2, CM - Manage Masks, and choosing option **2** to create a mask. Then while editing the new CM mask, you can cut and paste the contents of the old 7.1 mask file into the new CM mask file.

4.9.1 Field descriptions

There is a description of fields available, which you can use during masking. You first have to enter your masking profile with **E** , then you can see information about the parameters in the member header. See Figure 4-65.

```
DB2 Admin ------------------- CM - Masks -------------------------- Row 1 of 4
Command ===>                                                  Scroll ===> CSR

Line commands:
 U - Update  DEL - Delete  INS - Insert  ML - Mask lines  CH - Changes
 E - Edit  I - Details on mask

Sel          ID Owner    Name                Comment
              * *        *                   *
--- ---------- -------- ------------------ ----------------------------------
             63 PAOLOR1  PROT.DBAF.MASK.01
E            64 PAOLOR1  PROT.DBAF.MASK.02
             66 PAOLOR1  PROT.DBAF.MASK.03
             67 PAOLOR1  PROT.DBAF.MASK.04
****************************** END OF DB2 DATA ******************************
```

Figure 4-65 Choosing a mask for edit

After editing your mask entry, you jump into a temporary data set as shown in Figure 4-66. It provides you the mask syntax and tells you what fields you can change.

```
DB2 Admin --------------------  -------------------------- Columns 00001 00072
Command ===>                                                 Scroll ===> CSR

****** **************************** Top of Data ******************************
==MSG>
==MSG>  Mask Syntax:
==MSG>    field:inmask,outmask
==MSG>  Fields (hierarchy):
==MSG>    COLNAME
==MSG>    NAME
==MSG>      DBNAME,TSNAME,TBNAME,IXNAME,UDFNAME,
==MSG>      UDTNAME,COLLNAME,PKGNAME,PGMNAME,PLNNAME
==MSG>      DBRMNAME,STPNAME,SFNAME,TGNAME,GRPNAME,
==MSG>      VCATNAME,GBPNAME
==MSG>      BPNAME
==MSG>        TSBPNAME,IXBPNAME
==MSG>      SGNAME
==MSG>        TSSGNAME,IXSGNAME
==MSG>    AUTHID
==MSG>      OWNER,SQLID,SCHEMA
==MSG>      GRANTID
==MSG>        GRANTOR,GRANTEE
==MSG>  Examples:
==MSG>    OWNER:ABC*,DEF*
==MSG>    NAME:PRE*,NPRE*
==MSG>
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001 OWNER   :DSG24D0,DSG24D01
000002 DBNAME  :DSG24D0,DSG24D01
000003 AUTHID  :DSG24D0,DSG24D01
****** *************************** Bottom of Data ***************************
```

Figure 4-66 Masking descriptions

Note: Although it looks like a data set, all entries are stored within Change Management.

4.9.2 Migrating an existing masking data set

If you already used the masking facility for Admin and Compare 7.1 and lower, you can easily migrate your masking data set to the table. Just enter **manage masks** from the Change Management Menu and enter an owner and a name for your masking entry. Then go back to the previous menu and display the created mask. Although your masking entry is not a data set, it appears to be one if you try to open this data set for EDIT with the line command **E**.

Then you can easily copy your existing masking data set to your new masking table entry. This is very helpful because many users of Admin and Compare use masking data sets with a lot of entries. After copying, press PF3 and all of your entries are stored as rows into your masking table :cmown.ADBCMASKS automatically.

Note: Make sure that you use UNNUM in your existing masking data set before you copy your masking data set to your table in order to prevent the numerations from being stored as well.

4.9.3 Updating a masking entry in your Change Management database

If there are changes not complete yet and you want to update your masking entry — for instance, renaming your entry — then you are prompted that updating your masking might be dangerous because errors might occur. See Figure 4-67.

```
DB2 Admin -- DB8A CM - Warning: Dependent Changes ---------------- 20:03

Updating a mask is dangerous because errors might occur while
creating a delta for target, or importing a delta. Continue
only if you are sure updating the mask is safe.


Select a choice
  1.  Cancel
  2.  Continue
  3.  Show dependent changes
```

Figure 4-67 A warning pop-up when updating a mask being used in a non-complete change

Note: We do not recommend updating a mask while there are changes not yet complete.

4.9.4 Masking during Compare

Notice the difference between masking while importing a delta change data set and masking while comparing objects respectively. In both cases, you want to apply your naming convention:

- When you use Compare masking, you want to apply them before you do the Compare.
- When you use masking while importing a delta change data set, you do your naming convention adoption without the scope of comparing.

4.9.5 SAVE handling for a mask

Whenever you update a mask, make sure that you point-and-shoot to SAVE before leaving the panel. This guarantees that all your updates have been saved.

4.10 Using the ignore feature

The ignore feature allows you to prevent some of the parameters for your objects from being compared. For instance, you know that there are differences in terms of PQTY and SQTY between two table spaces. If you *ignore* those fields, then Compare does not report them, although Compare might have detected a difference between your table space objects in terms of PQTY and SQTY.

Ignores can be saved in the same way in Change Management tables as your masking definition.

4.10.1 Updating the ignored fields

You can show the ignore lines in your ignore name with **IL** and update them with **U**. You can then select all the fields you want to exclude from comparing. See Figure 4-68.

```
DB2 Admin ------------------ CM - Ignores ------------------------- Row 1 of 3
Command ===>                                                  Scroll ===> CSR
INSERT stmt executed
Line commands:
 U - Update  DEL - Delete  INS - Insert  IL - Ignore lines  CH - Changes
 I - Details on ignore

Sel          ID Owner    Name                Comment
              * *        *                   *
--- ----------- -------- ------------------ ----------------------------------
              1 PAOLOR2  EXCLUDE CREATOR
             21 PAOLOR2  NULL VERSION       FOR COMPARING TO NULL FILE
IL           41 PAOLOR1  PROT.DBAF.IGNO.01  MASTERIGNORE
****************************** END OF DB2 DATA ******************************
```

Figure 4-68 Showing the ignore lines

You then can update the ignore fields with the command **U**. In our example in Figure 4-69, we want to ignore specific fields from SYSTRIGGERS.

Archived

```
Compare ---------------- Specify Ignore Fields: Objects ---------- Row 1 of 18
Command ===>                                                  Scroll ===> CSR

Valid line commands are:
 U - Update Ignore Fields

Select Object          Ignore Fields
       *               *
------ --------------- -------------------------------------------------------
       SYSDATABASE     CREATOR
       SYSINDEXPART    PQTY,SQTY
       SYSTABLEPART    PQTY,SQTY,COMPRESS
       SYSTABLESPACE   CREATOR,SEGSIZE,DSSIZE
U      SYSTRIGGERS     None
       GENERIC         None
       SYSCHECKS       None
       SYSCOLUMNS      None
       SYSDATATYPES    None
       SYSFIELDS       None
       SYSINDEXES      None
       SYSKEYS         None
       SYSPARMS        None
       SYSRELS         None
       SYSROUTINES     None
       SYSSEQUENCES    None
       SYSTABLES       None
       SYSVIEWS        None
****************************** END OF DB2 DATA ******************************
```

Figure 4-69 Update an ignore field from SYSIBM.SYSTRIGGERS

We choose the field OWNER by selecting it. It is added to the ignored fields for SYSTRIGGERS. See Figure 4-70.

```
Compare --------------- Select Ignore Fields for SYSTRIGGERS ------ Row 1 of 7
Command ===>                                                  Scroll ===> CSR

Valid line commands are:
 S - Select (add) field  U - Un-select

Select Fields            Action
       *                 *
------ ----------------- --------
s      OWNER
       TRIGTIME
       TRIGEVENT
       GRANULARITY
       TEXT
       REMARKS
       TRIGNAME
****************************** END OF DB2 DATA ******************************
```

Figure 4-70 Preventing the trigger OWNER from comparing

Switch back with PF3 to the previous panel and you can see that the owner has been added to SYSTRIGGERS. See Figure 4-71.

```
 ---------------- Specify Ignore Fields: Objects ----------------- Row 1 of 18
Command ===>                                                   Scroll ===> CSR

Valid line commands are:
 U - Update Ignore Fields

Select Object          Ignore Fields
       *               *
------ --------------- ---------------------------------------------------------
       SYSDATABASE     CREATOR
       SYSINDEXPART    PQTY,SQTY
       SYSTABLEPART    PQTY,SQTY,COMPRESS
       SYSTABLESPACE   CREATOR,SEGSIZE,DSSIZE
       GENERIC         None
       SYSCHECKS       None
       SYSCOLUMNS      None
       SYSDATATYPES    None
       SYSFIELDS       None
       SYSINDEXES      None
       SYSKEYS         None
       SYSPARMS        None
       SYSRELS         None
       SYSROUTINES     None
       SYSSEQUENCES    None
       SYSTABLES       None
*      SYSTRIGGERS     OWNER
       SYSVIEWS        None
****************************** END OF DB2 DATA ******************************
```

Figure 4-71 OWNER has been added to SYSTRIGGERS for ignoring

4.10.2 SAVE handling for an ignore

Whenever you update an ignore, make sure that you point-and-shoot to SAVE before leaving the panel. This guarantees that all your updates have been saved.

Archived

4.11 Change terminology

This section should help you understand the different terminology that is used when you work with Change Management. We describe this wording in more detail.

4.11.1 Generating a first change

Before discussing the terminology, we show you some screen captures of how to generate a first change. This is a prerequisite for basic understanding. To keep it easy, we start by using the ALC function of Admin.

If you plan to recreate this scenario, make sure that your SQLID is mentioned as OPTIONAL in the ID table, as you learned in 2.4, “DB2 Security for Change Management” on page 39.

Let us start (with ALC) on a table to do an update of a column description. See Figure 4-72.

```
DB2 Admin --------------- DB8A Tables, Views, and Aliases -------- Row 1 of 11
Command ===>                                                  Scroll ===> CSR

Commands: GRANT  MIG
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
 V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
 ? - Show all line commands

Sel   Name                Owner    T DB Name  TS Name    Cols        Rows Checks
      *                   *        * *        *            *           *      *
----- ------------------ -------- - -------- -------- ------ ----------- ------
ALC   EMP                 DSN8810  T DSN8D81A DSN8S81E     14          42      2
      EEMP                DSN8810  T DSN8D81A DSN8S81R     16          -1      0
      EPROJ               DSN8810  T DSN8D81A DSN8S81R     10          -1      0
      EACT                DSN8810  T DSN8D81A DSN8S81R      5          -1      0
      EPROJACT            DSN8810  T DSN8D81A DSN8S81R      7          -1      0
      EEPA                DSN8810  T DSN8D81A DSN8S81R      8          -1      0
      PROJ                DSN8810  T DSN8D81A DSN8S81P      8          20      0
      ACT                 DSN8810  T DSN8D81A DSN8S81P      3          18      0
      PROJACT             DSN8810  T DSN8D81A DSN8S81P      5          77      0
      PARTS               DSN8810  T DSN8D81A DSN8S81S      4          -1      0
      EMPPROJACT          DSN8810  T DSN8D81A DSN8S81P      6          74      0
****************************** END OF DB2 DATA ******************************
```

Figure 4-72 Using ALC on a DSN8 sample table

Archived

Because the SQLID, PAOLOR1, is registered inside the ID table as OPTIONAL, we are asked if we want to use Change Management for this ALC process. You should enter YES to confirm working with Change Management. See Figure 4-73.

```
DB2 Admin ----------- DB8A Change Management Prompt ----------- 14:18

Change Management is optional for SQLID:  PAOLOR1

Do you wish to use Change Management for this function: YES (Yes/No)
```

Figure 4-73 Asking for Change Management

We just want to do a very easy update. As you can see, we increase the column definition SEX from 1 to 2 bytes. After pressing <ENTER>, we point-and-shoot to CONTINUE. See Figure 4-74.

```
DB2 Admin ------------------- DB8A ALTER Table ------------------- Row 1 of 14
Command ===>                                                  Scroll ===> CSR

New owner ===> DSN8810  >                    Old owner : DSN8810
New name  ===> EMP                >          Old name  : EMP
Volatile  ===> NO        Action    : DROP-SC  New DB ===> DSN8D81A
Rows/Page    : 37.236    Partitions: 5        New TS ===> DSN8S81E
Commands :  CONTINUE  PRIMKEY  ADD  REL  ALTPART
Line commands :
 I - Insert  U - Update  D - Delete  R - Repeat  LAB - Label  COM - Comment
 M - Move    A - After   B - Before  X - Index   RES - Reset update
                                                              Old Operation
Select Column Name        Col No Col Type Length  Scale Null D Col No Type
       *                   * *            *      * *    *      * *
------ ----------------> ------ -------- ------ ------ ---- - ------ ---------

       EMPNO                 1 CHAR          6      0 N    N      1
       FIRSTNME              2 VARCHAR      12      0 N    N      2
       MIDINIT               3 CHAR          1      0 N    N      3
       LASTNAME              4 VARCHAR      15      0 N    N      4
       WORKDEPT              5 CHAR          3      0 Y    Y      5
       PHONENO               6 CHAR          4      0 Y    Y      6
       HIREDATE              7 DATE          4      0 Y    Y      7
       JOB                   8 CHAR          8      0 Y    Y      8
       EDLEVEL               9 SMALLINT      2      0 Y    Y      9
*      SEX                  10 CHAR          2      0 Y    Y     10 UPDATE
       BIRTHDATE            11 DATE          4      0 Y    Y     11
       SALARY               12 DECIMAL       9      2 Y    Y     12
       BONUS                13 DECIMAL       9      2 Y    Y     13
       COMM                 14 DECIMAL       9      2 Y    Y     14
****************************** END OF DB2 DATA ******************************
```

Figure 4-74 Using ALC

Archived

We are now moving to the Change Management registration panel. See Figure 4-75. By default, the owner of the change is set to the value of the SQLID (PAOLOR1 in this case). We just have to enter a change name. The product automatically fills in a delta version owner and name. A delta version is created during change registration and contains an internal representation of your change. Versions in general are described later in 4.12, “Versions” on page 186.

```
DB2 Admin -------------------- CM - Register Options -------------------- 14:37
Option ===>

Commands: CONTINUE                                      DB2 System: DB8A
                                                        DB2 SQL ID: PAOLOR1
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR1  >  (Optional, default is PAOLOR1)
Name  . . . . . . . . D070307 - 1ST TEST CHANGE                          >
Comment . . . . . . .                                                    >
Change Type . . . . : CHANGE   (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                      Owner         Name
Ignore  . . . . . . .          >                          >  (? to look up)
Mask  . . . . . . . .          >                          >  (? to look up)
Delta Version . . . . PAOLOR1  >    D070307 - 1ST TEST CHAN >  (? to look up)
```

Figure 4-75 Enter your change

You then must point-and-shoot to CONTINUE, your first change becomes registered. You can see a confirmation in your panel, `Register Successful`. See Figure 4-76.

```
DB2 Admin -------------------- CM - Register Options -------------------- 14:48
Option ===>
Register Successful
Commands: CONTINUE                                      DB2 System: DB8A
                                                        DB2 SQL ID: PAOLOR1
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR1  >  (Optional, default is PAOLOR1)
Name  . . . . . . . . D070307 - 1ST TEST CHANGE                          >
Comment . . . . . . .                                                    >
Change Type . . . . : CHANGE   (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                      Owner         Name
Ignore  . . . . . . .          >                          >  (? to look up)
Mask  . . . . . . . .          >                          >  (? to look up)
Delta Version . . . . PAOLOR1  >    D070307 - 1ST TEST CHAN >  (? to look up)
```

Figure 4-76 Register successful

There is a short command that you can use from almost every panel to go to the Change Management menu. The command is **CMM**; you use this command as a primary command in the command line.

In Figure 4-77, Figure 4-78, and Figure 4-79, you can see your change previously registered.

Enter **1** - Manage changes first. See Figure 4-77.

```
DB2 Admin -------------------- Change Management (CM) ------------------- 15:43
Option ===> 1


   1 - Manage changes                                    DB2 System: DB8A
   2 - Manage masks                                      DB2 SQL ID: PAOLOR1
   3 - Manage ignores
   4 - Manage versions
   5 - Manage ID table
   6 - Report changes
```

Figure 4-77 Manage changes

Enter **1** - Display changes. See Figure 4-78.

```
DB2 Admin ---------------------  CM - Manage Changes -------------------- 15:45
Option ===> 1


   1 - Display changes                                   DB2 System: DB8A
   2 - Create a change                                   DB2 SQL ID: PAOLOR1
   3 - Create delta for target
   4 - Import changes









Enter display selection criteria (Using a LIKE operator, criteria saved):
Name . . . . . . . D070307%          >  Created by . .          >
Owner  . . . . . . PAOL%    >           Altered by . .          >
Type . . . . . . .                      Status . . . .
Created within . .                      Change ID  . .
Altered within . .
```

Figure 4-78 Display changes

Archived

Enter **1** - Display changes, and put the change name you recently entered in the field's Name and Owner. See Figure 4-79.

```
DB2 Admin ------------------ CM - Changes ------------------------- Row 1 of 1
Command ===>                                                   Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- -----------------> ------- -------- ---------------------
          83 PAOLOR1  D070307 - 1ST TEST CHANGE  DEFINED
****************************** END OF DB2 DATA ******************************
```

Figure 4-79 Change Panel according to your query

From Figure 4-79, you can then see an overview of your changes according to your query from the previous panel. In our case, just the one change we entered matches.

Owner and name are already known from Figure 4-75, where we entered the change. But there is a new important column: ID. This column is an internal unique identifier set by Change Management. You cannot modify this value with Change Management, it is automatically set by an identity column definition.

The status column indicates that our change has a status of DEFINED. It has been successfully registered, but has yet to be analyzed or run. The columns Status and Type are described respectively in Table 4-5 on page 162 and Table 4-6 on page 163.

4.11.2 Pending change

To show you what a pending change means and how a pending change is detected, let us go quickly through the same panel flow.

We try to create a second change on the same table and the same column with the same command as we did already in Figure 4-72 on page 172. As you know from Figure 4-73 on page 173 you are asked again if you plan to control the upcoming change with Change Management or not. Once more we answer with YES.

Instead of the "expected" ALC panel, another panel is shown which lets you know about an *existing change related to the object where you want to perform an action against.* This is what we call a pending change. In our example, it is the change which was previously created in Figure 4-74 on page 173. The name of the pending change is shown here in Figure 4-80.

The relationship is, by current design, that they share the same database.

```
DB2 Admin ------------ Pending Changes - Conflict Resolution ------ Row 1 of 1
Command ===>                                                   Scroll ===> CSR

Commands: CONTINUE
Line commands:
 CH - Change  I - Interpret

Pending changes exist for objects
Apply virtual changes . . .            (Apply, Supersede)

Sel Owner    Name                 Statement
    *        *                    *
--- -------- -----------------> ---------------------------------------------->
    PAOLOR1  D070307 - 1ST TEST  ADMIN ALTER TABLE "DSN8810"."EMP"    ALTER COL
****************************** END OF DB2 DATA ******************************
```

Figure 4-80 Detecting an existing pending change

In general, a pending change is a change that is going to be executed, but the DDL has not been applied yet to your DB2 subsystem. The change might or might not be in an ANALYZED status.

Figure 4-81 provides an overview of how Change Management detects pending changes. Pending changes are detected before the change is registered.

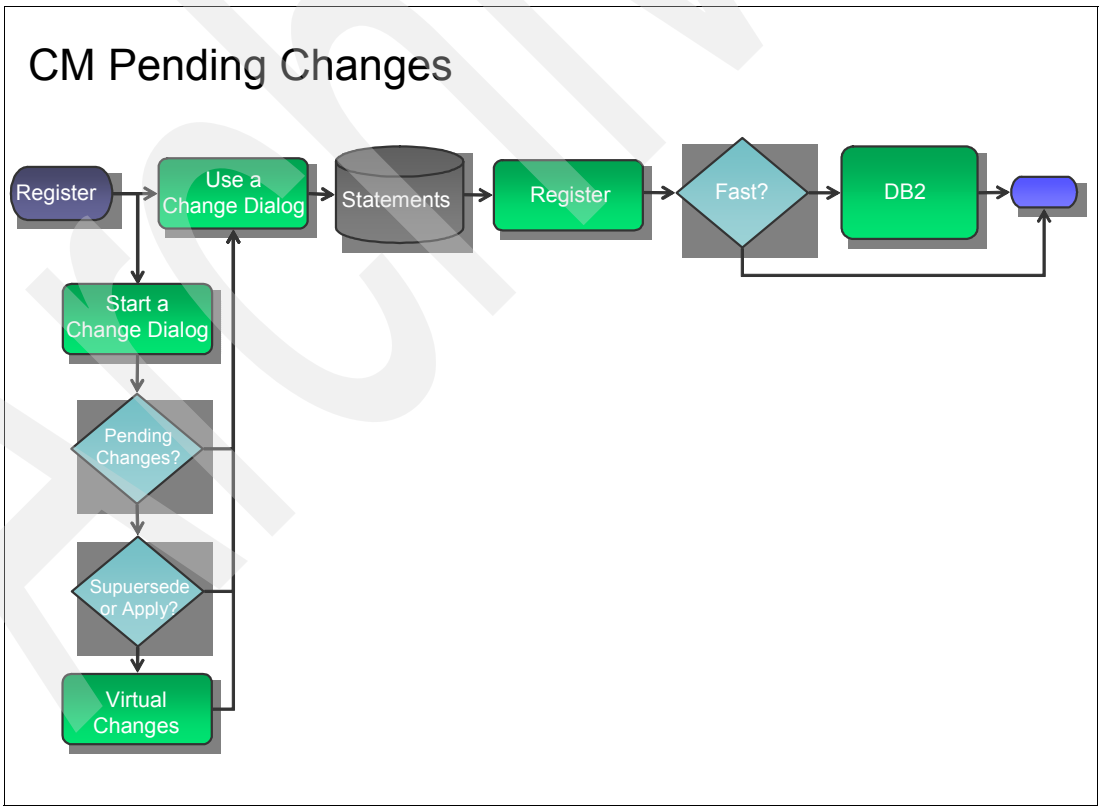

Figure 4-81 Diagram Detecting Pending Changes

Notice that the process of “detecting a pending change” currently works on a database level. So even if you try to access another table in another table space of the same database, you get the same panel that informs you about the fact that a pending change exists.

You might also see pending changes (on IMPORT, COMPARE, immediate changes) when objects share the same storage group.

Once you are informed about pending changes, you are currently not informed about the common object they share. Two commands are very helpful to determine the common resource of the pending changes.

You can enter the command **CH** in front of your change to see the complete change entry. See Figure 4-82.

```
DB2 Admin ------------ Pending Changes - Conflict Resolution ----- Row 1 of 14
Command ===>                                                  Scroll ===> CSR

Commands: CONTINUE
Line commands:
 CH - Change  I - Interpret

Pending changes exist for objects
Apply virtual changes . . .          (Apply, Supersede)

Sel Owner    Name                 Statement
    *        *                    *
--- -------- ------------------ ---------------------------------------------->
CH  PAOLOR1  D070312-CR#34-NOR  DROP TABLE "DSN8810"."EEMP"
    PAOLOR1  D070312-CR#35-NOR  DROP TABLE "DSN8810"."EMP"
    PAOLOR1  D070313-CR#55-NOR  ALTER TABLESPACE DSN8D81A.DSN8S81S PRIQTY 47 SE
    PAOLOR1  D070313-CR#55-NOR  ALTER TABLESPACE DSN8D81A.DSN8S81S PRIQTY 48 SE
    PAOLOR1  D070315-CR#59-NOR  ALTER TABLESPACE DSN8D81A.DSN8S81S PRIQTY 720 S
    PAOLOR1  D070315-CR#60-NOR  ALTER TABLESPACE DSN8D81A.DSN8S81S PRIQTY 48  S
    PAOLOR1  D070315-CR#63-NOR  ALTER TABLESPACE DSN8D81A.DSN8S81S PRIQTY 720 S
    PAOLOR1  D070315-CR#66-NOR  ALTER TABLESPACE DSN8D81A.DSN8S81S PRIQTY 720 S
    PAOLOR1  D070316-CR#88-NOR  ADMIN DROP TABLE DSN8810.EMP
    PAOLOR1  D070316-CR#88-NOR  CREATE TABLE DSN8810.EMP (EMPNO CHAR(6) FOR SBC
    PAOLOR1  D070316-CR#88-NOR  GRANT DELETE,INSERT,SELECT,UPDATE ON TABLE DSN8
    PAOLOR1  D070316-CR#90-NOR  ADMIN DROP TABLE DSN8810.EMP
    PAOLOR1  D070316-CR#90-NOR  CREATE TABLE DSN8810.EMP(EMPNO CHAR( 6) FOR SBC
    PAOLOR1  D070316-CR#90-NOR  GRANT DELETE, INSERT, SELECT, UPDATE ON TABLE D
****************************** END OF DB2 DATA ******************************
```

Figure 4-82 Enter the command CH to display your change entry

Then you can enter the command **SR** to see the related objects to this change (Figure 4-83).

```
DB2 Admin ------------------ CM - Changes ------------------------- Row 1 of 1
Command ===>                                                   Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- --------------------
sr       143 PAOLOR1  D070312-CR#34-NOR  CHANGE  DEFINED
****************************** END OF DB2 DATA ******************************
```

Figure 4-83 Enter the command SR to see the related objects

From the current product design, determining the common object of a pending change and the change you want to enter is manual work. In Figure 4-84, you can see the related objects of the pending change from Figure 4-83.

```
DB2 Admin -- Browse Result of SQL Select              ------------ Row 1 of 3
Command ===>                                                   Scroll ===> CSR

L CHANGEID   CHANGESEQ OBJECTTYPE QUALIFIER NAME     BASEIND
         *           * *         *         *            *
- -------- ---------- ---------- --------- -------- -------
       143          1 DB                   DSN8D81A       0
       143          1 TB         DSN8810   EEMP           1
       143          1 TS         DSN8D81A  DSN8S81R       0
****************************** END OF DB2 DATA ******************************
```

Figure 4-84 The related objects

After you are notified about a pending change, you must tell Change Management how to proceed by indicating how the pending changes should be handled. There are two choices: APPLY and SUPERSEDE.

APPLY

Apply means that you apply these changes as just *virtual* and not real. The purpose is to work with your new change on a basis that shows you the pending changes as applied, but keep in mind that in fact they are *not* applied yet to DB2. This is what we call a virtual change. You can build up your new change based on this virtual change.

Currently there are some restrictions when using APPLY, and your pending changes have become virtual:

- If a pending change exists as a CREATE INDEX, you cannot apply an additional index as a virtual change. So if you type the REL command to see the relating objects, you can find out that the “pending” index is not shown on the list.
- In the same way, new views are not applied as a virtual change.

The scope while applying the pending changes can be currently at database level or at storage group level. For that reason, you might experience wait time if you apply all your related pending changes. The more unfinished changes (DEFINED, ANALYZED) you have, the more time it takes until the Apply is finished. Currently not only those changes listed at a pending change are applied, but all unfinished changes are applied as a virtual change. This can make APPLY time-consuming if you have many unfinished changes.

SUPERSEDE

Supersede means that you want to proceed with your planned activity without applying your pending changes as a virtual change. The superseded change becomes a prerequisite for all the pending changes it has superseded. In addition, those pending changes go back to status DEFINED.

You can use the abbreviation **A** for APPLY or **S** for SUPERSEDE. After typing **S** or **A**, point-and-shoot to CONTINUE.

4.11.3 Further actions against a pending change

You can perform some minor actions, for instance, you can interpret the change. You can get more information by typing **I** as a line command. See Figure 4-85 for how this looks.

```
 DB2 Admin -------- CM - Interpretation of a Change Statement ----------- 12:24
 Command ===>

 Details for change sequence 1
 of change ID 83

 Change ID  . . . . . : 83          Statement type . . : ALTER
 Change sequence  . . : 1           Object type  . . . : Table
 Object qualifier . . : DSN8810     Object name  . . . : EMP
 Last altered . . . . : 2007-03-07-14.48.22.712501
 Statement  . . . . . :
 ADMIN ALTER TABLE "DSN8810"."EMP"    ALTER COLUMN "SEX"     CHAR  (2) FOR SBCS
DATA WITH DEFAULT NULL
```

Figure 4-85 Change interpretation

4.11.4 Virtual change

A virtual change is a change that is registered and has not yet been executed. The status of such a change can be DEFINED or ANALYZED. A virtual change gives you the opportunity to enter a new change based on pending activities (pending changes) to be done later. A virtual change can be created when Change Management detects a pending change and you reply with APPLY. Refer to the description of a pending change in 4.11.2, “Pending change” on page 176.

4.11.5 Emergency change

Both changes, immediate and emergency, are proposed to be executed fast. They are both fast changes. The difference between an immediate change and an emergency change is related to the existence of pending changes. In case of an emergency change, the pending changes which are in status ANALYZED go back to status DEFINED. Both changes do not have an ANALYZE step and they are supposed to be executed without an **RN** command.

For instance, you want to add a new column to an existing table with a DB2 Alter command. If there are no pending changes known on the database level, you are asked to execute this “fast” change as an immediate change. Otherwise, you are asked to execute it as an emergency change.

We start with the Admin functionality to execute SQL statements from screen input. You can invoke this screen as shown in Figure 4-86.

```
DB2 Admin -------------- DB2 Administration Menu 7.2.0 ------------------ 13:12
Option ===> 2.1

   1 - DB2 system catalog                                DB2 System: DB8A
   2 - Execute SQL statements                            DB2 SQL ID: PAOLOR1
   3 - DB2 performance queries                           Userid    : PAOLOR1
   4 - Change current SQL ID                             DB2 Rel   : 815
   5 - Utility generation using LISTDEFs and TEMPLATEs
   P - Change DB2 Admin parameters
  DD - Distributed DB2 systems
   E - Explain
   Z - DB2 system administration
  SM - Space management functions
   W - Manage work statement lists
   X - Exit DB2 Admin
  CC - DB2 catalog copy version maintenance
  CM - Change management

Interface to other DB2 products and offerings:
   C  DB2 Object Comparison Tool
   I  DB2I
  WL  Workload Generator
```

Figure 4-86 Execute SQL

Figure 4-87 shows panel, the Execute SQL Statements from Screen Input.

```
DB2 Admin ----------- Execute SQL Statements from Screen Input ---------- 20:59
Command ===>

                                                       DB2 System: DB8A
SQL statement:                                         DB2 SQL ID: PAOLOR1
ALTER TABLE DSN8810.EEMP ADD LASTCOL CHAR(1) NOT NULL WITH DEFAULT;
```

Figure 4-87 Execute SQL statements from screen input

Archived

You are asked if you want to use Change Management because the SQLID, PAOLOR1, has been registered as OPTIONAL. See Figure 4-88.

```
DB2 Admin ----------- DB8A Change Management Prompt ----------- 13:20
Top of data
Change Management is optional for SQLID: PAOLOR1

Do you wish to use Change Management for the statement: y   (Yes/No)

                                                        More:     +
Statement that is about to be executed (first 28 lines)
ALTER TABLE DSN8810.EEMP ADD LASTCOL CHAR(1) NOT NULL WITH DEFAULT
```

Figure 4-88 Change Management

This time we make a change to the table DSN8810.EEMP. In our explanation of pending changes in 4.11.2, “Pending change” on page 176, we used table DSN8810.EMP. Both tables belong to the database DSN8D81A. Because pending changes are registered internally on a database level, you are informed that there are pending changes. See Figure 4-89.

```
DB2 Admin ------------------ DB8A CM - Register Change ------------------ 13:26
Option ===>

 C - Cancel
 E - Register and run as an emergency change
 N - Register as a normal change, pending changes become prereqs
 S - Register as a normal change, supersede pending changes
 D - Display pending changes to the same object(s)

For option E enter the following information for the change:
 Owner  . . . . .          >  (Optional, default is PAOLOR1)
 Name . . . . . .                  >
 Comment  . . . .                                                      >
                                                            More:     +
Statement that is about to be executed (first 28 lines)
ALTER TABLE DSN8810.EEMP ADD LASTCOL CHAR(1) NOT NULL WITH DEFAULT


 There are pending changes related to the objects you are modifying.
 Use the "Display pending changes" option to see the pending changes.
```

Figure 4-89 Possibility to run an emergency change

Archived

You can show them with option **D** and then switch back to the register panel in Figure 4-89. See Figure 4-90.

```
DB2 Admin ------------------ DB8A CM - Pending Changes ------------ Row 1 of 2
Command ===>                                                   Scroll ===> CSR

Line commands:
 CH - Change  I - Interpret

Sel Owner    Name                Statement
    *        *                   *
--- -------- ------------------ ---------------------------------------------->
    PAOLOR1  D070308-CR#27-NOR   ADMIN ALTER TABLE "DSN8810"."EMP"    ALTER COL
    PAOLOR1  D070308-CR#28-NOR   ADMIN ALTER TABLE "DSN8810"."EMP"    ALTER COL
****************************** END OF DB2 DATA ******************************
```

Figure 4-90 Showing pending changes before doing an emergency change

As soon as you type **E**, your change is executed. All the pending changes shown above are superseded immediately. Notice that superseding the changes puts them back in a defined status if you analyzed them already.

After doing the emergency change, you can see that it is done in a prompt. See Figure 4-91.

```
DB2 Admin ----------- Execute SQL Statements from Screen Input ---------- 15:33
Command ===>
ALTER stmt executed
                                                   DB2 System: DB8A
SQL statement:                                     DB2 SQL ID: PAOLOR1
ALTER TABLE DSN8810.EEMP ADD LASTCOL CHAR(1) NOT NULL WITH DEFAULT;



Press ENTER to execute the SQL statement, or enter EDIT on the command line to
edit it. Use command CAPS (OFF/ON) to change the caps mode.
```

Figure 4-91 Emergency change has been executed

In the Changes menu you can see that your emergency (fast) change has been executed and there the status COMPLETE has been set.

Archived

4.11.6 Immediate change

An immediate change is a fast change as well. You can do an immediate change if there are no pending changes. Otherwise, it is an emergency change as described in 4.11.5, “Emergency change” on page 180.

You try to add a column to the table DSN8810.EEMP. See Figure 4-92.

```
DB2 Admin ----------- Execute SQL Statements from Screen Input ---------- 16:13
Command ===>

                                                        DB2 System: DB8A
SQL statement:                                          DB2 SQL ID: PAOLOR1
ALTER TABLE DSN8810.EEMP ADD LASTCOL CHAR(1) NOT NULL WITH DEFAULT;
```

Figure 4-92 Adding a column

You are asked to use Change Management. See Figure 4-93.

```
DB2 Admin ----------- DB8A Change Management Prompt ----------- 16:16

Change Management is optional for SQLID: PAOLOR1

Do you wish to use Change Management for the statement: y   (Yes/No)

                                                       More:     +
Statement that is about to be executed (first 28 lines)
ALTER TABLE DSN8810.EEMP ADD LASTCOL CHAR(1) NOT NULL WITH DEFAULT
```

Figure 4-93 Change Management

This time, because there is no pending change, you are not asked to execute your change as an emergency change. Therefore you now can run your change immediately. See Figure 4-94

```
DB2 Admin ------------------ DB8A CM - Register Change ------------------ 16:20
Option ===> I

 C - Cancel
 I - Register and run as an immediate change
 N - Register as a normal change



For option I enter the following information for the change:
 Owner  . . . . . PAOLOR1  >  (Optional, default is PAOLOR1)
 Name . . . . . . D070309-CR#33-IMM  >
 Comment  . . . .                                                        >
                                                                More:     +
Statement that is about to be executed (first 28 lines)
ALTER TABLE DSN8810.EEMP ADD LASTCOL CHAR(1) NOT NULL WITH DEFAULT
```

Figure 4-94 Register change

As soon as it is done, you are prompted so. You can look at the Change menu, where you might see that your (fast) immediate change ran successfully because of the COMPLETE status. See Figure 4-95.

```
DB2 Admin ------------------ CM - Changes ----------------------- Row 16 of 16
Command ===>                                                   Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- ---------------------
         121 PAOLOR1  D070309-CR#33-IMM  FAST    COMPLETE
****************************** END OF DB2 DATA ******************************
```

Figure 4-95 Fast change is completed

Note: If your change fails, you might see the status, FAILED. There is no way of restarting a failed emergency or immediate change. You have to set up a new change then.

4.11.7 Prerequisite change

There are some situations where a change depends on another change. Those changes are simply called *prereq changes*. You can easily see if a change has prerequisite changes or not by typing **PQ** in front of a change. The prereq changes are displayed.

In the panel in Figure 4-96, we enter **PQ** as a line command before a change.

```
DB2 Admin ------------------ CM - Changes ------------------------- Row 2 of 2
Command ===>                                                   Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- ---------------------
PQ       143 PAOLOR1  D070312-CR#34-NOR  CHANGE  ANALYZED
****************************** END OF DB2 DATA ******************************
```

Figure 4-96 Look up for prerequisite changes

Archived

Next, the CM - Prerequisites panel is shown in Figure 4-97.

```
DB2 Admin --------------- CM - Prerequisites ---------------------- Row 1 of 1
Command ===>                                                   Scroll ===> CSR

Prerequisites for change "PAOLOR1"."D070312-CR#34-NOR"
Line commands:
 U - Update  DEL - Delete  INS - Insert  CH - Show change
 I - Details on prerequisite

Sel Owner    Name                Comment
    *        *                   *
--- -------- ------------------ ----------------------------------------------
    PAOLOR1  D070312-CR#33-NOR
****************************** END OF DB2 DATA ******************************
```

Figure 4-97 Prereq changes can easily be determined by typing PQ

Note: A pending change does not necessarily have to be a prerequisite change.

4.11.8 Delta change

A delta change is the task that you do when you perform the PROMOTE process. Remember, during PROMOTE you compare two base versions. The data set produced from a delta change is called a *delta change data set*. How to use the PROMOTE process is described in 4.4.5, “The PROMOTE process” on page 151.

4.12 Versions

A version is a snapshot of object definitions at a given point of time. It is used by many DB2 Admin and DB2 Compare functions that invoke a compare process.

Versions can be created and used in multiple ways:

- When using CM, you can manually generate a version based on a version scope (Base Version).
- When you use DB2 Compare, then there are always versions generated. This is regardless of the compare source, whether it is DB2 catalog or DDL input — the product always generates a version and then compares the versions.
- During the REGISTER process, the product builds a delta version.
- During the analyze process, the product works with various types of version files.
- When you run a change using CM, you can optionally specify to create a base version using a version scope after the changes have been applied.
- When you do promote, then you work on base versions. In this case, base versions are just used, not created.
- When Change Management is enabled, you have the option of storing versions in data sets or in the Change Management database.

It can be sometimes be confusing to distinguish the various types of versions and when and how they are used. Therefore we have tried to group the descriptions based on how you can control the usage. We came up with three different types:

- **Base:** Completely user driven:

  You completely control the usage.
- **Delta:** Completely product driven:

  You have no control of how this is used.
- **Target and source base:** Partially user driven and partially product driven:

  This is a mix of both — you specify a method and the product then does the rest.

We wanted to focus on whether you must be concerned about this version topic, or whether the product does it all behind the scenes and you have no way of influencing it.

4.12.1 Base

You have complete control regarding how to work with base versions:

- Version scope
- Base version
- Generating a base version

Tip: Consider storing all of your versions in the Change Management database. This makes them easier to track, access, and recover.

How to build a base version can be seen in Figure 4-98.

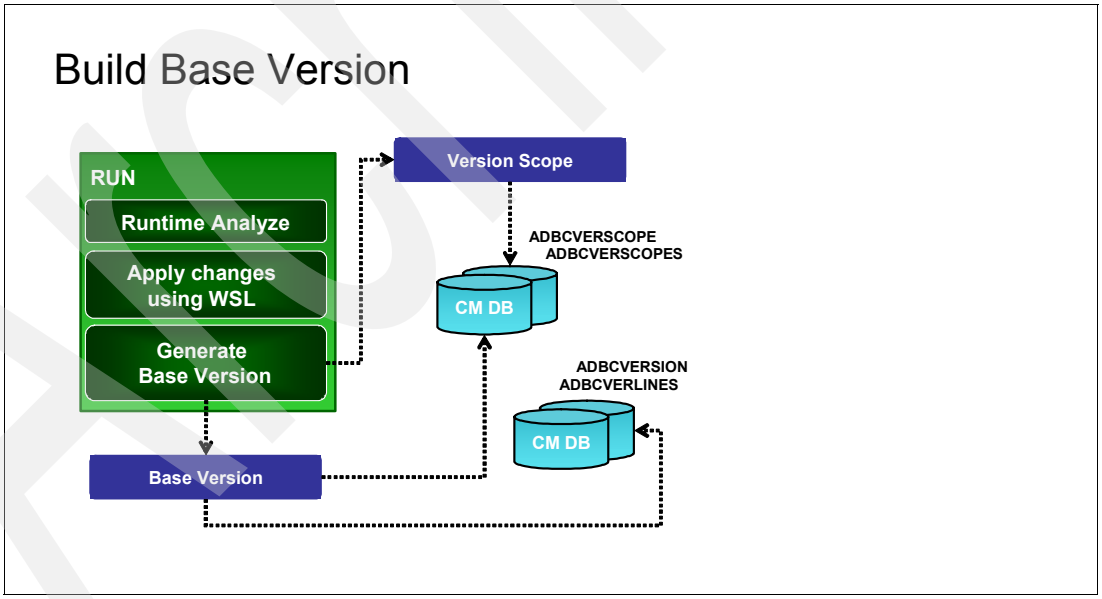

Figure 4-98 Build a base version during the RUN process

Version scope

Before we can start working with a base version, we have to tell the product for which objects we want to take a base version. This is done by defining a version scope and later the base versions refer to that version scope. The version scope determines the objects that are included in a base version.

A version scope can be any set of objects, such as one or more databases, or a group of table spaces. Typically, you want to define scopes that identify all of the objects for an application or application area. For example, the scope for a human resources application should contain all the human resource databases.

Maintaining a version scope is a manual process, and you should ensure that the definition of the scope always includes all of the objects that you intended. Also, you might not want to change the scope because this can lead to undesired results later when comparing base versions that were generated based upon different scopes (meaning different object sets).

Each version scope has a ScopeID, an Owner, and a Name.

Creating a version scope

In this section we describe how to define a version scope. On the Change Management (CM) panel, as shown in Figure 4-99, we select option **4**.

```
DB2 Admin -------------------- Change Management (CM) ------------------- 19:45
Option ===> 4


   1 - Manage changes                                  DB2 System: DB8A
   2 - Manage masks                                    DB2 SQL ID: PAOLOR3
   3 - Manage ignores
   4 - Manage versions
   5 - Manage ID table
   6 - Report changes
```

Figure 4-99 CM - Manage versions option

As we want to create a version scope, we specify option **3** - Insert a version scope, on the Manage Versions panel as shown in Figure 4-100. The Insert/Update panel allows us to either insert or update one of several types of objects. The supported types of objects are masks, ignores, and version scopes.

```
DB2 Admin ------------------- CM - Manage Versions ---------------------- 20:02
Option ===> 3


   1 - Display versions                                DB2 System: DB8A
   2 - Display version scopes                          DB2 SQL ID: PAOLOR3
   3 - Insert a version scope


Enter display selection criteria (Using a LIKE operator, criteria saved):
Name . . . . . .                    >  Created by . .          >
Owner  . . . . . .          >          Altered by . .          >
Created within . .                     Version ID . .
Altered within . .
```

Figure 4-100 CM - Manage Versions panel

We specify as version scope name DSG24D0% and enter a comment which is optional. When pressing Enter, the new version scope is inserted in the CM table ADBCVERSCOPE. See Figure 4-101.

```
DB2 Admin ---------------- CM - Insert Version Scope -------------------- 19:58
Command ===>
INSERT stmt executed

Owner . . . PAOLOR3  >  (Optional, default is PAOLOR3)
Name  . . . DSG24DO%       >  (Required, ? to look up)
Comment . . Scope all databases DSG24DO%               >
```

Figure 4-101 CM - Insert Version Scope

After pressing F3 we return to the Manage Versions panel shown in Figure 4-100. As the next step, we want to include objects to our new version scope. Therefore we select option **2** to display the version scopes as shown in Figure 4-102.

```
DB2 Admin ------------------- CM - Manage Versions ---------------------- 14:06
Option ===> 2


   1 - Display versions                                DB2 System: DB8A
   2 - Display version scopes                          DB2 SQL ID: PAOLOR3
   3 - Insert a version scope









Enter display selection criteria (Using a LIKE operator, criteria saved):
Name . . . . . . . DSG24DO%          >  Created by . .          >
Owner  . . . . . .         >           Altered by . .          >
Created within . .                     Version ID . .
Altered within . .
```

Figure 4-102 Manage Versions panel - Display version scopes

We specify the **SO** line command to insert objects in the version scope that we created. The Version Scope Objects panel is displayed, as shown in Figure 4-103.

```
DB2 Admin --------------- CM - Version Scopes ---------------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR

Line commands:
 VE - Versions  SO - Version scope objects  GV - Generate new base version
 INS - Insert  U - Update  DEL - Delete  I - Details on version scope

Sel          ID Owner    Name                Comment
              * *        *                   *
--- ----------- -------- ------------------ ----------------------------------
SO            3 PAOLOR3  DSG24DO%           SCOPE ALL DATABASES DSG24DO%
****************************** END OF DB2 DATA ******************************
```

Figure 4-103 CM - Version Scopes

We can now add each object that we want to have in the version scope. Example 4-2 shows the values to specify in the TYPE, QUAL, and NAME keywords for each type of object.

Example 4-2 The keyword values of the request parameters for each object type

Object Type	TYPE	QUAL	NAME	Notes
Database	DB	n/a	*dbanme*	
Table space	TS	dbname	*tsname*	
Table	TB	creator	*tbname*	
View	VW	creator	*vwname*	
Alias	AL	creator	*aliasname*	
Index	IX	creator	*ixname*	
User-defined data type	DT	schema	*udtname*	
User-defined function	FU	schema	*udfname*	
Stored procedure	SP	schema	*stpname*	
Sequence	SQ	schema	*seqname*	For DB2 Version 8 and higher
Schema	SC	schema	n/a	
Trigger	TG	schema	*tgname*	
Storage group	SG	n/a	*sgname*	
Synonym	SY	creator	*syname*	

The values for the qualifier and name can contain zero or more of the following wildcard characters:

- A minus sign (-) represents any single character.
- A percent sign (%) or asterisk (*) represents one or more characters.
- Any other character represents a single occurrence of itself.

In Figure 4-104 we want to add all databases beginning with DSG24D0% in the version scope. As we entered object type **DB**, the value for the qualifier of the object is blank. This definition includes all underlying objects in the database (like table space, table, view, etc.) We do not have to specify every single object (although we could have).

```
DB2 Admin ----------- CM - Version Scope Objects ------------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR

Version scope objects for scope "PAOLOR3"."DSG24D0%"
Commands: SAVE
Line commands:
  I - Insert  D - Delete  R - Repeat

Sel T  Qual     Name               Oper.
    *  *        *                  *
--- -- -------> -----------------> ------

*   DB          DSG24D0%
****************************** END OF DB2 DATA ******************************
```

Figure 4-104 Example of version scope definition

We issue the **SAVE** primary command to insert the definition of the scope into the CM table ADBCVERSCOPES.

Base version

The base version is the snapshot of DB2 structures (not data) at a given time. This enables you to refer later to that instance of the definition. You can use it later in Compare processes to either undo changes or to rollout changes and refer to base versions as starting and ending point.

Creating a base version from a version scope

After we have created a version scope, we can now create a base version for that set of objects.

A version scope must exist if you plan to create a new base version when you apply changes. If you have a new base version created when you run a change to reflect the object definitions after the changes, you must specify the version scope for the version.

To generate a version based on that scope, we specify the **GV** line command on the Version Scopes panel as shown in Figure 4-105.

```
DB2 Admin --------------- CM - Version Scopes ---------------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR

Line commands:
 VE - Versions  SO - Version scope objects  GV - Generate new base version
 INS - Insert  U - Update  DEL - Delete  I - Details on version scope

Sel          ID Owner    Name               Comment
              * *        *                  *
--- ----------- -------- ----------------- ---------------------------------
GV            3 PAOLOR3  DSG24D0%          SCOPE ALL DATABASES DSG24D0%
****************************** END OF DB2 DATA ******************************
```

Figure 4-105 CM - Generate version

The ISPF pop-up panel Specify a New Base Version appears, as shown in Figure 4-106. Here we can specify an owner and name for the new version. We want to create a base version for a set of objects contained within version scope PAOLOR3.DSG24D0%.

```
DB2 Admin ------ Specify a New Base Version -------------------- 19:16 e
Command ===>                                                         e
                                                                     e
Enter/verify the following:                                          e
Owner  . . PAOLOR3  >  (? to look up)                                e
Name . . . DSG24D0% V0        >  (? to look up)                      e
```

Figure 4-106 CM - New base version

The JCL to create the version is displayed, as shown in Figure 4-107 and Figure 4-108. We submit the job to create the new version.

```
EDIT       PAOLOR3.SC63.SPFTEMP1.CNTL                      Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 //PAOLOR3# JOB (999,POK),'ITSO CM RESIDENCY',
000002 //         REGION=0M,MSGCLASS=X,CLASS=A,
000003 //         MSGLEVEL=(1,1),NOTIFY=&SYSUID
000004 /*JOBPARM S=SC63
000005 //*
000006 //        IF (RC>7) THEN
000007 //        ELSE
000008 //*********************************************************************
000009 //*
000010 //* STEP EXSCP1: EXTRACT VERSION SCOPE FROM DB
000011 //*
000012 //*******************************************************ADBCVSX*********
000013 //EXSCP1   EXEC PGM=IKJEFT01,DYNAMNBR=100
000014 //STEPLIB  DD DISP=SHR,DSN=GOC.V7R2M0.SGOCLLIB
000015 //         DD DISP=SHR,DSN=ADB.V7R2M0.SADBLLIB
000016 //         DD DISP=SHR,DSN=DB8A8.SDSNEXIT
000017 //         DD DISP=SHR,DSN=DB8A8.SDSNLOAD
000018 //SYSTSPRT DD SYSOUT=*
000019 //SYSTSIN  DD *
000020   DSN SYSTEM(DB8A)
000021   RUN PROG(ADBCVSX) PLAN(A72)
000022 //SYSPRINT DD SYSOUT=*
000023 //SCPOUT   DD DISP=(NEW,PASS,DELETE),
000024 //            SPACE=(TRK,(10,10,0)),LRECL=80,RECFM=FB
000025 //REPORT   DD SYSOUT=*
000026 //PARMIN   DD *
000027 SQUAL='PAOLOR3',
000028 SNAME='DSG24D0%';
000029 /*
000030 //        ENDIF
000031 //*************************************************************ADB2GEN***
000032 //        IF (RC>7) THEN
000033 //        ELSE
000034 //* STEP GNSQL2: GENERATE SQL FROM DB2 CATALOG
000035 //*********************************************************************
000036 //GNSQL2   EXEC PGM=IKJEFT01,DYNAMNBR=100
000037 //STEPLIB  DD DISP=SHR,DSN=GOC.V7R2M0.SGOCLLIB
000038 //         DD DISP=SHR,DSN=ADB.V7R2M0.SADBLLIB
000039 //         DD DISP=SHR,DSN=DB8A8.SDSNEXIT
000040 //         DD DISP=SHR,DSN=DB8A8.SDSNLOAD
000041 //SYSTSPRT DD SYSOUT=*
000042 //SYSTSIN  DD *
000043   DSN SYSTEM(DB8A)
000044   RUN PROG(ADB2GEN) PLAN(A72) -
000045   PARMS('/WRTCAT,NOGEN,REBIND')
000046   END
000047 /*
```

Figure 4-107 JCL to create Base Version (1/2)

See Figure 4-108 for the continuation of the job listing.

```
000048 //SYSPRINT DD SYSOUT=*
000049 //SQL      DD DUMMY
000050 //SQLOUT   DD DUMMY
000051 //CAT      DD DSN=&&CATOUT,
000052 //            DCB=(LRECL=16800,RECFM=VB,DSORG=PS),
000053 //            SPACE=(TRK,(15,15),RLSE),
000054 //            UNIT=SYSDA,
000055 //            DISP=(NEW,PASS)
000056 //IN       DD DISP=SHR,
000057 //            DSN=*.EXSCP1.SCPOUT
000058 //        ENDIF
000059 //*******************************************************************
000060 //*
000061 //* STEP IMVER3:
000062 //*
000063 //*****************************************************ADBCVIM*********
000064 //        IF (RC>7) THEN
000065 //        ELSE
000066 //IMVER3   EXEC PGM=IKJEFT01,DYNAMNBR=100
000067 //STEPLIB  DD DISP=SHR,DSN=GOC.V7R2M0.SGOCLLIB
000068 //         DD DISP=SHR,DSN=ADB.V7R2M0.SADBLLIB
000069 //         DD DISP=SHR,DSN=DB8A8.SDSNEXIT
000070 //         DD DISP=SHR,DSN=DB8A8.SDSNLOAD
000071 //SYSTSPRT DD SYSOUT=*
000072 //SYSTSIN  DD *
000073   DSN SYSTEM(DB8A)
000074   RUN PROG(ADBCVIM) PLAN(A72)
000075 //SYSPRINT DD SYSOUT=*
000076 //REPORT   DD SYSOUT=*
000077 //INDD     DD DSN=&&CATOUT,
000078 //            DISP=(OLD,DELETE)
000079 //PARMIN   DD *
000080 SQUAL='PAOLOR3',
000081 SNAME='DS24D0%',
000082 VQUAL='PAOLOR3',
000083 VNAME='DSG24D0% V0',
000084 CHANGEID=,
000085 VTYPE='BASE',
000086 ACTION='REPLACE';
000087 /*
000088 //        ENDIF
****** **************************** Bottom of Data ****************************
```

Figure 4-108 JCL to create Base Version (2/2)

After the base version has been generated, we can look it up with **VE** and see our base version (see Figure 4-109).

```
DB2 Admin ------------------ CM - Versions ------------------------ Row 1 of 1
Command ===>                                                  Scroll ===> CSR

Line commands:
 CH - Changes  PR - Promote  VS - Version scope  DEL - Delete  U - Update
 PT - Toggle protected status  I - Details on version

Sel          ID T Owner    Name                Comment
              * * *        *                   *
--- ----------- - -------- ------------------ -------------------------------
            266 B PAOLOR3  DSG24D0% V0
```

Figure 4-109 Versions panel showing a base version

So we have type **B** (for base) and we could also protect this base version (with option **PT**). This protection helps if we generate another base version (V1) and we accidentally specify the same name we had for V0.

Starting base version and ending base version

A starting base version represents the DDL before a change has been applied. Conversely, an ending base version represents the DDL after a change has been applied. Those two versions are used during a PROMOTE when you want to identify the differences (or the delta) between starting and ending base version. Let us put it mathematically as described in Example 4-3.

Example 4-3 DDL delta

```
abs( Ending base version - starting base version ) = DDL delta
```

4.12.2 Delta version

This is completely under control of the product, you cannot specify how it is used. The product uses delta versions internally during the REGISTER and ANALYZE processes. Figure 4-110 shows an example of how a delta version is created and stored during the REGISTER process.

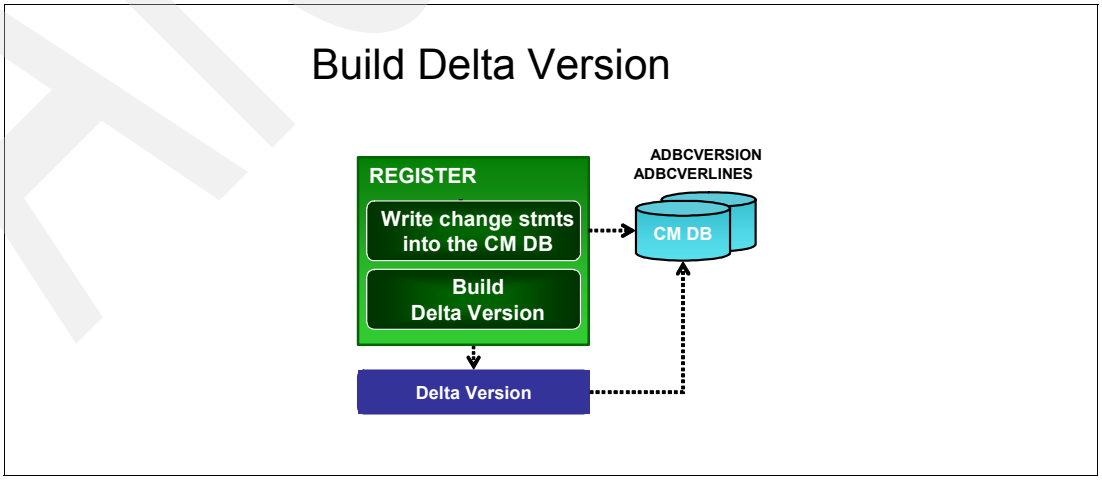

Figure 4-110 Build and store delta versions in CM database during REGISTER process

A delta version is automatically being created when you perform a change registration. In other words, your change statements being registered become an internal format stored in a version called a delta version. Because it is a delta version, it is a type D version. A delta version is stored in the Change Management tables :cmown.ADBCVERSION and :cmown.ADBCVERLINES.

Using versions in the ANALYZE process

This is partially under your control and partially controlled by the product in the ANALYZE process. You specify the base version method — and the product then executes its analysis based on your definition.

There are actually various versions in the ANALYZE process as shown in Figure 4-111.

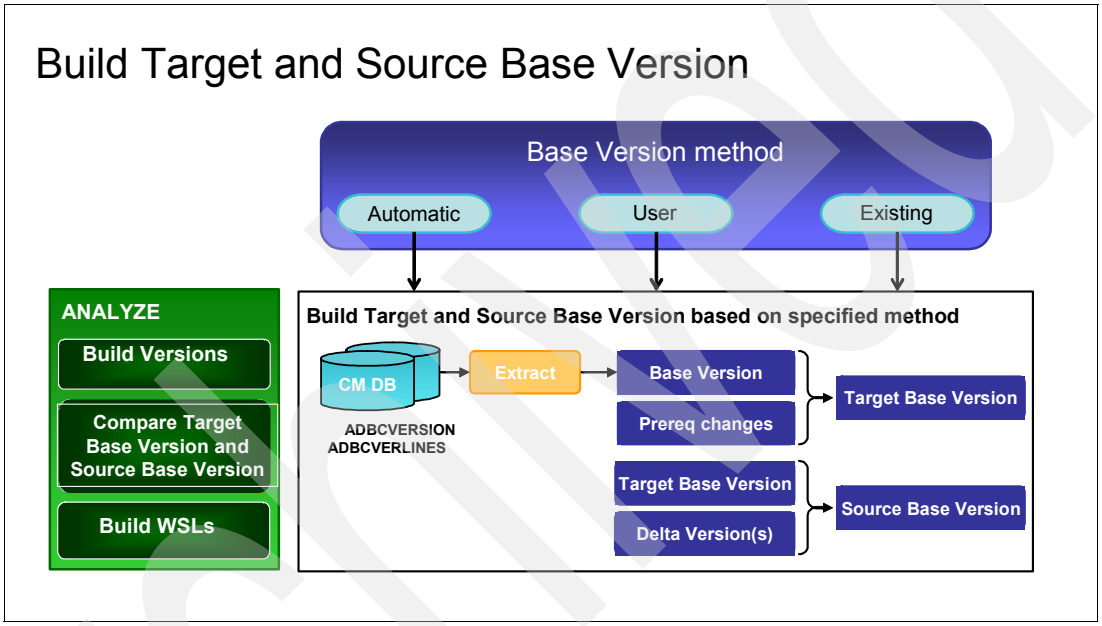

Figure 4-111 Use of versions in the ANALYZE process

Part of the ANALYZE process is to work from a base version. Any prerequisite changes are applied to the base version and this serves as the target for comparing. The Source for DB2 Compare is the target plus the changes for the change being analyzed. The base version can be established using one of the following methods.

- **Automatic**: CM automatically determines the object information to extract from the DB2 catalog which is then used as the base version.
- **User**: You specify a version scope that defines a list of object names that is used to extract the object information from the DB2 catalog which is then used as the base version.
- **Existing:** You specify an existing base version that exists in the CM database to use as the base version.

Target base version and source base versions

Those base versions are to be used just for internal purposes during the ANALYZE process. They are just temporary and are supposed to be deleted after the ANALYZE process. You cannot choose whether they are built or not but you can choose how they are built. Further explanation is available at 4.4.3, “The ANALYZE process” on page 139.

4.12.3 Versions: The whole picture

In Figure 4-112 we show a complete overview of CM versions.

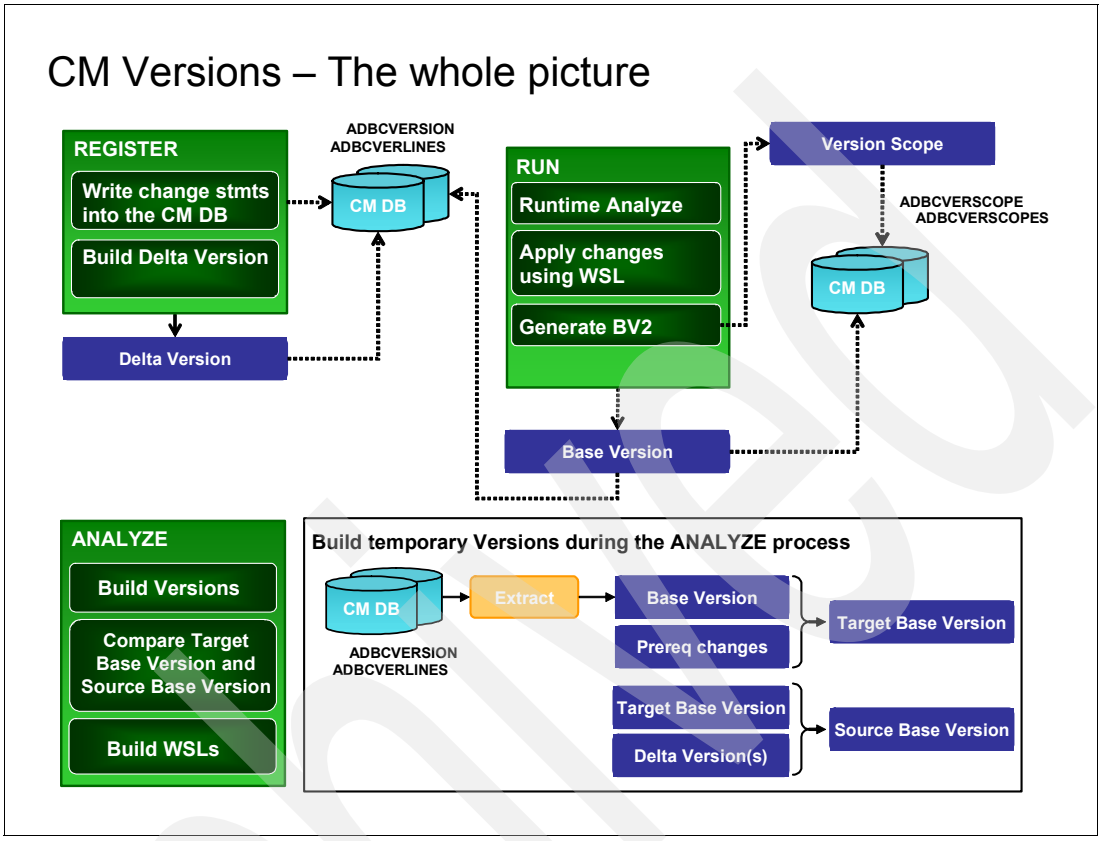

Figure 4-112 CM Versions - The whole picture

4.13 Change flow with Change Management

In this section we assume that we have to define and promote changes from a SOURCE system to a TARGET system. We show the flow of two possible scenarios when implementing the change using the CM functions. In the first scenario we use COMPARE, in the second one we use ALTER ALC.

4.13.1 Change on the source environment using COMPARE

In this scenario a base version (V0) is optional. To identify the differences between DB2 Catalog and the DDL, we generate a delta file from the COMPARE process.

REGISTER/ANALYZE/RUN

To be in sync with the target environment, we execute the change on the source environment. Therefore we store the change in the CM database (REGISTER), and go through ANALYZE and RUN.

Importing the changes on the target environment

We now implement the changes on the target environment. See the whole process in Figure 4-113.

1. IMPORT:

   We IMPORT the delta file into a change, and register the imported change in the Change Management database.

2. ANALYZE:

   We perform the ANALYZE process to generate a WSL.

3. RUN:

   We execute (RUN) the imported change using the generated WSL and capture a snapshot (Base Version Vn) of the objects after the change is complete.

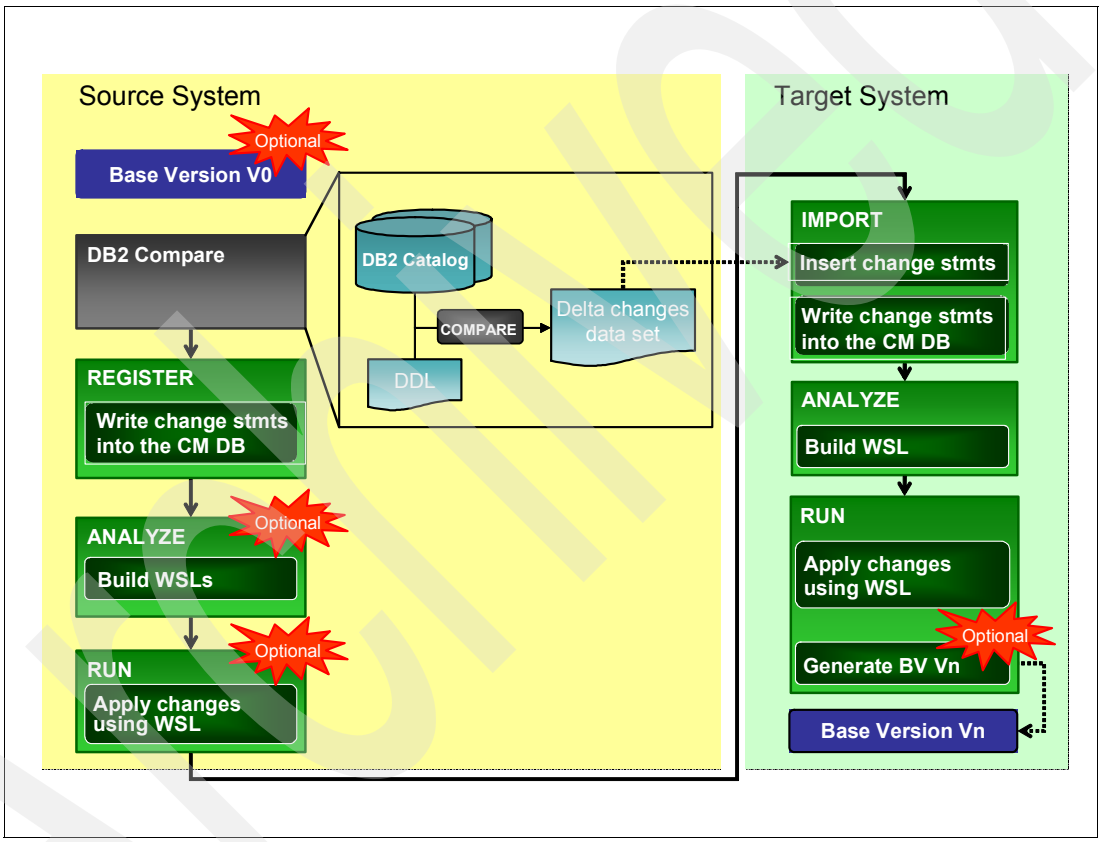

Figure 4-113 Implementing changes and rolling out new version (COMPARE)

4.13.2 Changes on the source environment using the ALTER ALC function

In this scenario a base version (V0) is a requirement to have a starting point. The starting base version is a snapshot of the objects before changes are made. During *promote* we have to compare the V0 with the ending version V1 to generate a delta changes data set that contains the SQL statements that represent the differences between V0 and V1.

1. REGISTER:

   After we have defined the change, we REGISTER the change in the Change Management database. A delta version that contains a catalog representation of the changes is stored in the CM database.

2. ANALYZE:

   The change must be verified (ANALYZE) and a WSL to run the change has been created.

3. RUN:

   Apply the changes by running (RUN) the generated WSL.

We want to have a new base version created after the changes have been made successfully.

Promoting the change to the target environment

To promote a change, two versions must exist. See Figure 4-114. During the promote process, DB2 Admin compares the ending version V1 with the starting version V0 and generates a delta changes data set. It contains the SQL statements that are required to bring the target system up to the same level as the source system.

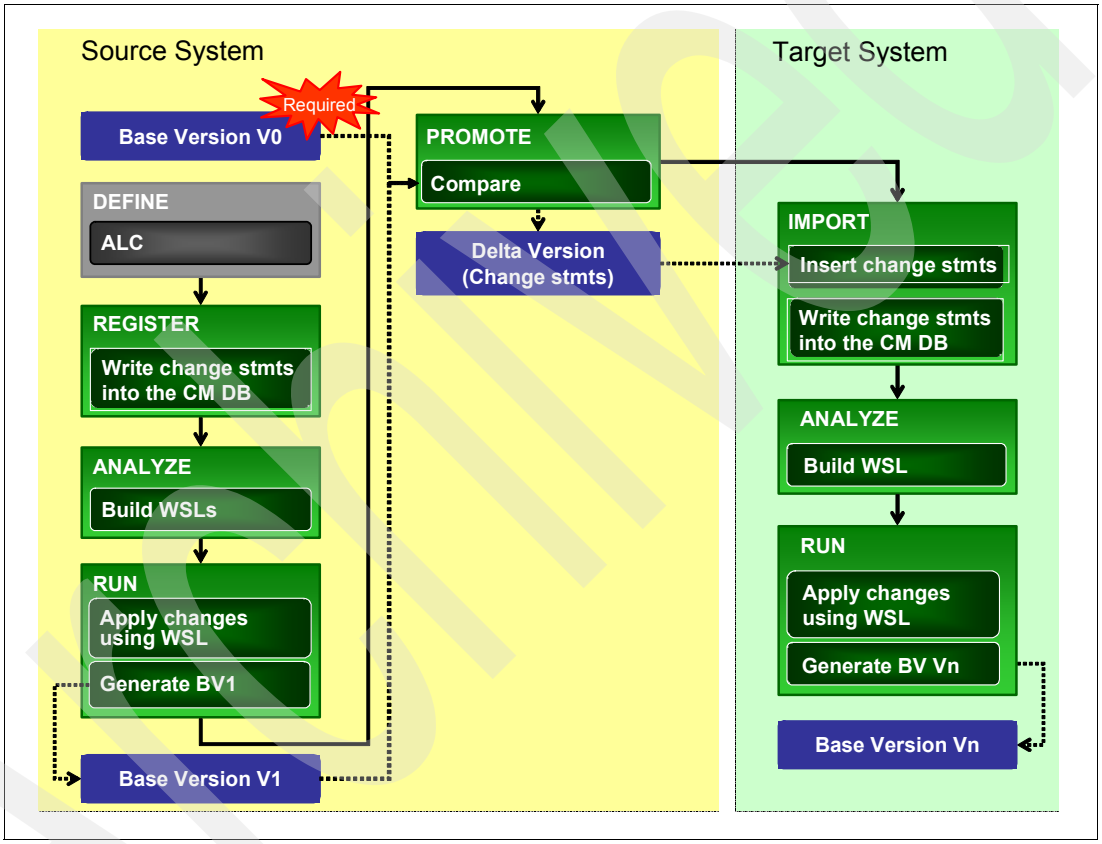

Figure 4-114 Implementing changes and rolling out new version (ALC)

Importing the changes on the target environment

The implementation of the changes on the target system is done through:

1. IMPORT:

   We now import the delta change file on the target system. During import a new change is created and registered.

2. ANALYZE:

   We analyze the change before executing.

3. RUN:

   We run the change and create a new base version Vn.

Part 3

Change Management scenario

In this part of the book, we give you an example of how change management works based on a defined scenario in an assumed customer landscape. Of course, every customer environment is different. Our intent is to guide you through the various phases of the change management process. We proceed by following these major steps:

1. Introduction of a sample IT database environment
2. Overview of the given change scenario
3. Implementation of an initial design (V0)
4. Roll-out of the initial design (V0) into other environments
5. Implementation of changes to the initial design (V1)
6. Roll-out of new versions of a database design (V1)
7. Recovery of changes

This part consists of the following chapters:

- Chapter 5, “Implementing an initial design (V0)” on page 201
- Chapter 6, “Implementing changes to a design (Vn)” on page 257
- Chapter 7, “Rolling out new versions of a database design” on page 311
- Chapter 8, “Recovery of changes” on page 329

© Copyright IBM Corp. 2007. All rights reserved.

Archived

5

Implementing an initial design (V0)

In this chapter, we first describe the sample IT database environment that we assume as a model for the implementation of our scenario. We then discuss different ways to implement an initial database design (Version 0). We make the distinction between:

- Implementing an initial design in the master development database (D0) environment *using DB2 Admin to generate and execute the DDL*
- Implementing an initial design in the master development database (D0) environment *using DDL that is generated externally (such as from a database design tool)*
- Rolling out the initial design to other environments

We have generated work statement lists (WSLs) in our scenarios working with the new CM functions.

© Copyright IBM Corp. 2007. All rights reserved.

5.1 Sample IT database environment

There are probably as many different IT database environments as there are IT shops. During this project, we defined a sample IT database environment (see Figure 5-1). While your environment might differ, hopefully you can still get some tips and ideas that are applicable to your environment. Of course, we realize that some modifications to our processes might be necessary in order to be applicable to your environment.

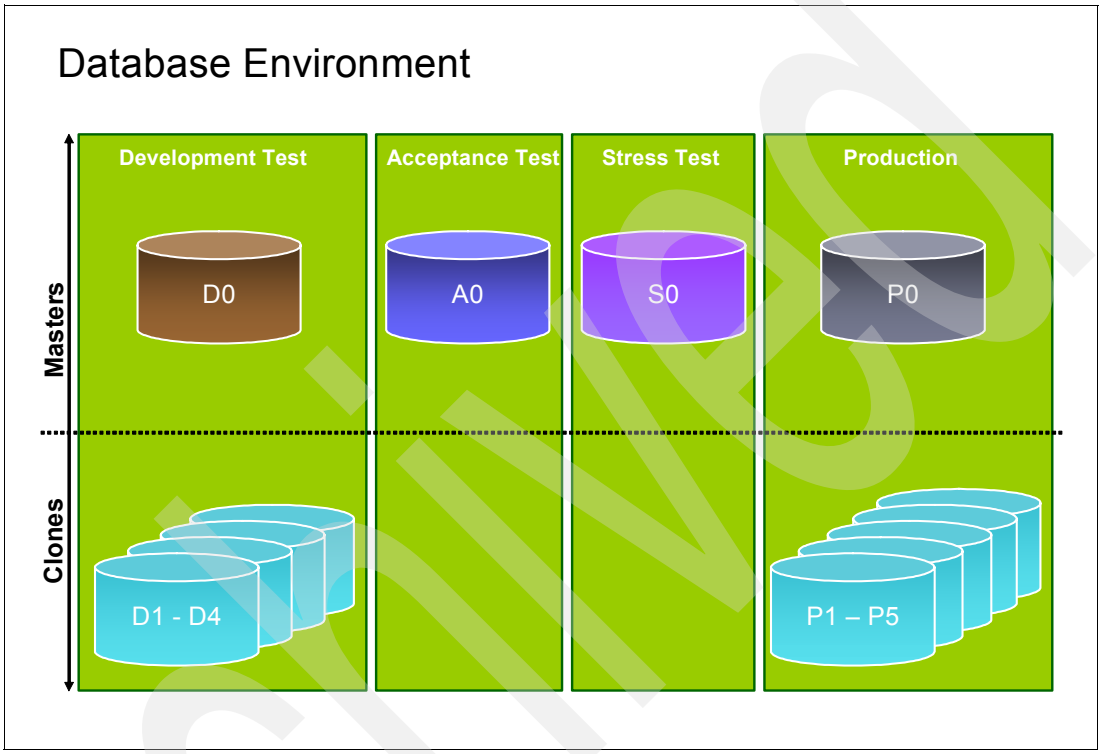

Figure 5-1 The database environment

Our sample logical database is called *SG24*. This is the basis for all naming conventions used in this book. This logical database name is the generic name for the database for a particular application. There can be many physical DB2 databases associated with a single logical database, especially if your shop's standard is to have one database per table space.

Here are the naming conventions we use for the sample environment for the SG24 database:

- **Database:**

  DSG24ecu

  Where:

  D Indicates that this is a database.
  SG24 Is our root name for our logical database.
  e Indicates the environment where the database is implemented:
  - *D* Development test
  - *A* Acceptance test
  - *S* Stress (volume) test
  - *P* Production

  c *0* For the master database implementation
  1-n For clones of the master database implementation
  u *1-4* For the user clones of the master database implementation

Attention: Since multiple people worked on this book simultaneously, each person was experimenting with their own copy of the database. Therefore, you can see that sometimes an additional character has been added to the names to avoid any naming conflicts. You can safely ignore that final character in the examples.

- **Owner:**

  DSG24ecu

  Our examples use the database root name, and then the environment as the owner, creator, or schema name.

  Tip: If it is desirable to reference more than one logical database in a program, then either an owner name that is indicative of the environment (that is, D0, D1, A0 and so on) and not the database name enables you to choose the environment at bind time. You can keep the owner name patterned after the database name if you created views that have the owner name linked to the environment.

- **Table space:**

  GLWSxxx

  Table space names in our environment always start with GLWS. The rest of the name is random.

5.1.1 Development test

Our sample development test environment consists of a master development database *DSG24D0 (D0)*, and four different database clones *(D1 - D4)* for different development groups. The DBAs are responsible for updating D0 with the latest version of the structural definition of the database, and populating the code tables. The DBAs are also responsible for cloning D0 to D1 - D4, as each application group is ready for the move to the next version of the database implementation. The individual application development groups are responsible for maintaining the actual test data necessary for their testing. In the development test environment, there is minimal data, and most object sizes are small.

5.1.2 Acceptance test

The acceptance test environment *(A0)* only has one master (D0). In this environment acceptance testing is done with a version from D0 that is planned to go to production. This environment might require more physical space to store data, but is intended for verification of the application's results in an integrated environment, not to test performance. The DBAs create the database structure by cloning a version of D0 along with populating the code tables, and the testers are responsible for populating the tables with test data.

5.1.3 Stress test

The stress test or volume test environment *(S0)* also has a single master. (We use the name "stress test" in this book to avoid confusion between "V" to represent the volume test environment and "V" to represent a certain version (level) of any environment.) In this environment performance testing is done with a version of the database structures from either D0 or A0. This environment should be sized and populated as close to the production environment as practical. The DBAs are responsible for cloning the database structure, populating the code tables, and sizing all the objects for the required volume. The test team is responsible for populating the tables with production-like data.

5.1.4 Production

The production environment consists of a master production database *(P0)* and five different customer databases *(P1 - P5)*. P0 could be an actual database, or could be files with all the DDL source and code table load files necessary to clone to a production site. Cloning from P0 to P1 - P5 is to be done with whatever version of P0 that the customer desires. The individual object sizes have to be estimated individually for each clone.

5.2 Developing DDL using DB2 Admin

There are multiple ways within DB2 Admin to create DB2 database objects. We do not describe every option in detail. We step through the GEN functionality that best fits our type of scenarios described in this book. Here are some of the possibilities:

- Create objects from scratch (with the CRE command), which is not described in detail in this book. This activity would trigger CM if activated, so that CREATEs can be fed into the CM process and get registered either as normal changes or as fast changes.
- Create objects from scratch with the DDL command on the system catalog panel (ADB21).
- Create new objects based on existing objects:
  - DDL works on a single object and does not generate dependent objects with it. This becomes apparent on the table level:
    - Foreign keys are not shown when DDL is issued against the table.
    - LOBs are not shown when DDL is issued against the base table.
  - GEN includes object dependencies that can be selected by the user.

5.2.1 Creating single objects (DDL)

Using the **DDL** line command, you can generate the DDL for a single object only. This function can be used for modeling DDL statements. For example, if you want the DDL for a certain table, you type in the **DDL** line command in front of that table. An example is described in 3.3.1, “Creating a limited number of objects (DDL)” on page 73.

We consider this process of generating DDL to be more the exception for cloning database environments, since it would be too tedious to generate all single DDL and add them all up in a WSL. DDL generation is useful if you want to use just that statement or want to only see the definition of the object.

5.2.2 Creating many objects based on existing objects (GEN)

DB2 Admin has a reverse engineering function (GEN) that lets you create objects and their dependencies from the DB2 system catalog.

You can, for example, create DB2 “Model” objects that meet your company’s standards. If you want to create a new object, you can use the model object as a basis to generate new objects. You can use *masking* with the GEN function, which enables you to easily change the model names into the real names that are required for the objects you want to create (as shown in Figure 5-2).

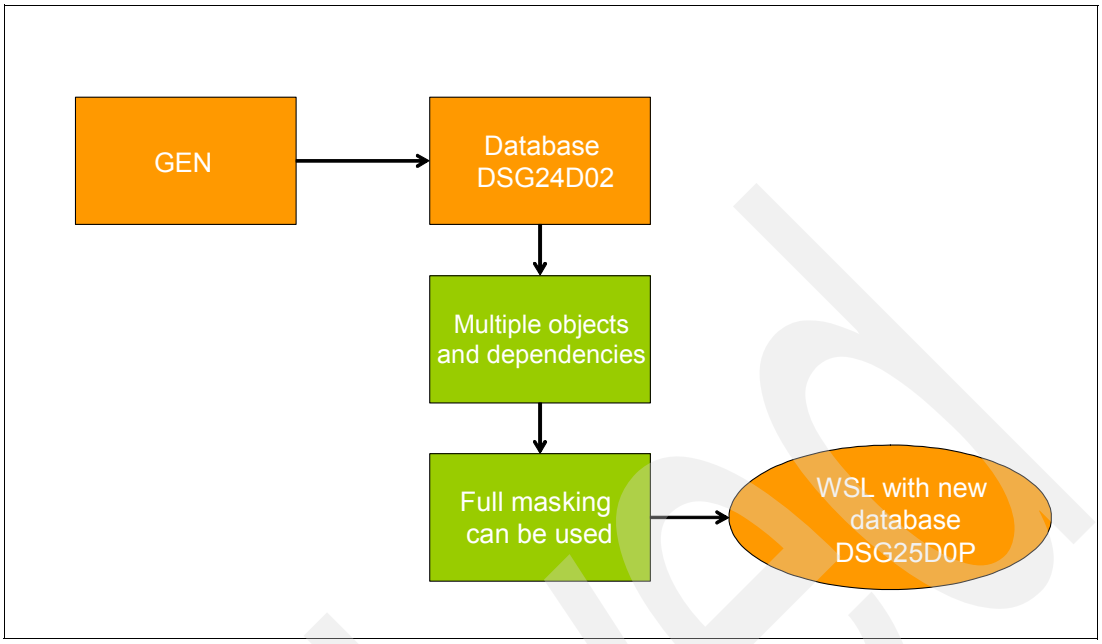

Figure 5-2 Creating database DSG25D0P based on existing database DSG24D02

We use the GEN function to create a database for a new project. The model for our generation is the development database “DSG24D02”. First, look up this database in the DB2 system catalog and enter the **GEN** line command, as shown in Figure 5-3.

```
 ADB21D in ------------------- DB8A Databases ---------------------- Row 1 of 7
Command ===>                                                  Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                          Storage  Buffer          Created      Index
Select Name     Owner     Group    Pool       DBID By       T E Buffer Pool
       *        *         *        *             * *        * * *
------ -------- -------- -------- -------- ------ -------- - - -----------
       DSG24D0  PAOLOR2  GLWG01   BP15        288 PAOLOR2    E BP16
       DSG24D01 PAOLOR2  GLWG01   BP15        294 PAOLOR2    E BP16
GEN    DSG24D02 PAOLOR2  GLWG01   BP15        295 PAOLOR2    E BP16
       DSG24D03 PAOLOR2  GLWG01   BP15        292 PAOLOR2    E BP16
       DSG24D04 PAOLOR2  GLWG01   BP15        296 PAOLOR2    E BP16
       DSG24D1  PAOLOR2  GLWG01   BP15        289 PAOLOR2    E BP16
       DSG24D2  PAOLOR2  GLWG01   BP15        293 PAOLOR2    E BP16
****************************** END OF DB2 DATA ******************************
```

Figure 5-3 Generating a database

Archived

This brings us to the GEN panel to generate SQL from DB2 catalog (see Figure 5-4).

```
 ADB2GEN n -------------- DB8A Generate SQL from DB2 catalog ------------- 21:29
Option ===>

 Generate SQL statements for database DSG24DO2          DB2 System: DB8A
                                                        DB2 SQL ID: PAOLOR2
                                                                More:   - +
   CREATE DATABASE. . . . . : Y (Y,N)   GRANT access ON DATABASE . : N (Y,N)
   CREATE TABLESPACE. . . . : Y (Y,N)   GRANT access ON TABLESPACE : N (Y,N)
   CREATE TABLE . . . . . . : Y (Y,N)   GRANT access ON TABLE. . . : N (Y,N)
   CREATE VIEW. . . . . . . : Y (Y,N,D) GRANT access ON VIEW . . . : N (Y,N)
   CREATE INDEX . . . . . . : Y (Y,N)   ALTER TABLE ADD FOREIGN KEY: D (Y,N,D)
   CREATE SYNONYM . . . . . : N (Y,N)   LABEL ON . . . . . . . . . : N (Y,N)
   CREATE ALIAS . . . . . . : N (Y,N)   COMMENT ON . . . . . . . . : N (Y,N)
   CREATE TRIGGER . . . . . : Y (Y,N,D) REBIND PLAN/PACKAGE. . . . : N (Y,N)
   CREATE STORAGE GROUP . . : N (Y,N)   GRANT use OF STORAGE GROUP : N (Y,N)

 New names/values for generated SQL: (leave blank to use current values)
   Object owner. . . . . : DSG25DOP >  Run SQLID. . . . . . . . . : <NONE>
   Object grantor. . . . :          >
   Alloc TS size as. . . : DEFINED     (DEFINED, USED, or ALLOC)
   Database name . . . . : DSG25DOP
   Storage group for TS. :          >  Storage group for IX . . . :         >
   Target DB2 version. . :             (Current DB2 version: 815)
   Use Masking . . . . . : YES            (Yes/No)
   Generate catalog stats: NO             (Yes/No/Only)
     Target cat qualifier:           >  (Default is SYSIBM)

 Output file and execution mode:
   Add to work stmt list : YES            (Yes/No)
   Data set name . . . . :
     Data set disposition: OLD            (OLD, SHR, or MOD)
   Execution mode. . . . : BATCH          (BATCH or TSO)
   Commit statements per :                (Db, tS, Tb, All, None. Default is All)
   DB2 defaults handling :                (Keep, or Remove. Default is Keep)

 DB2 Command output file:
   Data set name . . . . :                                                  >
     Data set disposition: OLD            (OLD, SHR, or MOD)

 BP - Change batch job parameters
```

Figure 5-4 Generating SQL for a database from the DB2 Catalog

On this panel we specify a **Y** for all objects we want to create using DB2 Admin.

For CREATE VIEW we use **D**, which indicates that views are extracted, but DB2 Admin does not check that all other objects referenced in the view, also are being generated. This significantly reduces the resource consumption to generate all DDL statements, especially when running against a large DB2 catalog.

For the ALTER TABLE ADD FOREIGN KEY option, we entered **D** to extract FOREIGN KEYS for tables that are dependent on the tables being extracted.

We also define the names that should be changed and choose the following values for other fields:

Object owner	Our new owner is **SG25DOP**.

Tip: You can also USE MASKING for translating the object owner.

Run SQLID	**<NONE>** is used to avoid SET CURRENT SQLID when creating objects.
Alloc TS size as	We want to use the **defined** size for the table spaces.
Database name	Our new database name is **DSG25DOP.**

Tip: You can also USE MASKING for translating the database name.

Storage group	We leave it blank and let GEN use for both table spaces and index spaces the storage group name used from the originating objects.
Use Masking	We specify **Y** because we want to change other names, such as table space names.

Tip: USE MASKING makes sense if there is a requirement for changes that are not covered on this panel. This panel only covers object owner, database name, and storage group name.

Add to work stmt list	We specify **Y** because we do want to run the DDL later from a work statement list.
Data set name	We can leave this blank as it is not used when we add the DDL to a work statement list.
Execution mode	As we prefer to run jobs in the background, we specify **BATCH** to generate a batch job for WSL generation.
Commit statements per	We use the default (All).
DB2 defaults handling	We use the default (Keep).
Data set name	We do not specify a command output data set, since we do not ask for REBIND statements to be generated.

After pressing the Enter key, DB2 Admin asks for the name of the masking file you want to use (see Figure 5-5). It can either be an entry in the CM database (ADBMASK table) or a sequential file.

If you specify an owner/name and not a mask DSN name, then masking is defined in the CM database. If you specify a mask DSN name, then masking is defined in a sequential file. If the file does not exist, DB2 Admin creates a new file and enables you to edit it.

If you have not activated CM, then DB2 Admin protects the mask table entry fields so that you cannot enter any data, and uses a sequential file.

In our example we are creating a mask to be stored in CM.

Archived

```
ADB2MSKP
EssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
e ADB2GENM  ----------------------- Specify Mask ---------------------------- e
e Command ===>                                                                e
e                                                                             e
e Mask Table Entry:                                                           e
e   Owner . . . . PAOLOR2  >            (? to look up)                        e
e   Name  . . . . DSG24DO2-DSG25DOP  >  (? to look up)                        e
e Data Set:                                                                   e
e  Mask DSN . . .                                                             e
e Options:                                                                    e
e  Edit Mask  . . YES                (Yes/No)                                 e
e                                                                             e
e                                                                             e
e                                                                             e
DsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

Figure 5-5 Specify Mask Data Set panel

Since we have to define a new mask, we first get prompted to specify a mask name and also a comment. Both values that we provide should be meaningful so that their purpose can be determined more easily later. See Figure 5-6.

```
ADB2C22 n ---------------- CM - Insert Mask ---------------------------- 20:10
Command ===>


Owner . . . PAOLOR2  >  (Optional, default is PAOLOR2)
Name  . . . DSG24DO2-DSG25DOP  >  (Required, ? to look up)
Comment . . Masking from Database DSG24DO2 to DSG25DOP  >
```

Figure 5-6 Insert Mask panel

After pressing Enter, the new entry into the mask table gets confirmed with the message, `INSERT stmt executed` (see Figure 5-7).

```
ADB2C22 n ---------------- CM - Insert Mask ---------------------------- 20:18
Command ===>
INSERT stmt executed

Owner . . . PAOLOR2  >  (Optional, default is PAOLOR2)
Name  . . . DSG24DO2-DSG25DOP  >  (Required, ? to look up)
Comment . . MASKING FROM DATABASE DSG24DO2 TO DSG25DOP  >
```

Figure 5-7 Execution of Insert Mask

Then we have to press PF3 to get out of it and go to the mask lines panel, which allows us to enter the mask entries.

Tip: If you do not PF3 out and press Enter instead, then DB2 Admin tries to insert this mask again and gives you an SQLCODE -803 (indicating that you already have the mask inserted). Then on every Enter, you get the SQLCODE -803. You have to PF3 out two times until you get to the CM mask lines panel.

We want to change the table space names from GLWSx to **TS25**x, and we specify this accordingly as shown in Figure 5-8.

```
ADB2C2L n ----------------- CM - Mask Lines ----------------------- Row 1 of 1
Command ===>                                                  Scroll ===> PAGE

Mask lines for mask "PAOLOR2"."DSG24DO2-DSG25DOP"
Commands: SAVE
Line commands:
  I - Insert  D - Delete  R - Repeat  M - Move  A - After  B - Before

Sel    Sequence Type     From                  To                    Oper.  T
              * *        *                     *                     *      *
--- ----------- -------- --------------------> ----------------------> ------ -

              1 TSNAME   GLWS*                 TS25*
****************************** END OF DB2 DATA ******************************
```

Figure 5-8 Masking entries (Type, From, and To) provided by the user

Tip: We have already defined the masking of DBNAME and OWNER on the GEN panel. We could have also added DBNAME and OWNER in the mask table (again), but we do not want to duplicate this effort.

Tip: Then we get the confirmation on the **UPDATE** in the Oper. column. And in order to store the row in the DB2 table, we have to actually issue the **SAVE** command (pressing Enter and PF3 is not enough). And then we get the confirmation that the insert succeeded (see Figure 5-9).

```
ADB2C2L n ----------------- CM - Mask Lines ----------------------- Row 1 of 1
Command ===> save                                             Scroll ===> PAGE

Mask lines for mask "PAOLOR2"."DSG24DO2-DSG25DOP"
Commands: SAVE
Line commands:
  I - Insert  D - Delete  R - Repeat  M - Move  A - After  B - Before

Sel    Sequence Type     From                  To                    Oper.  T
              * *        *                     *                     *      *
--- ----------- -------- --------------------> ----------------------> ------ -

*             1 TSNAME   GLWS*                 TS25*                 UPDATE
****************************** END OF DB2 DATA ******************************
```

Figure 5-9 CM Mask Lines entry panel

Archived

After pressing the PF3 key, you get prompted to provide a work statement list name as shown in Figure 5-10.

```
EsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
e DB2 Admin ------------------ Specify Work Statement List ----------------- e
e                                                                            e
e                                                                            e
e  Work stmt list dsn  . . . 'ADMIN.WSL'                                     e
e  Work stmt list name . . . DSG25D0P                                        e
e                                                                            e
e                                                                            e
e                                                                            e
e                                                                            e
e                                                                            e
e                                                                            e
e                                                                            e
e                                                                            e
DsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

Figure 5-10 Specify Work Statement List prompt panel

Then you press Enter and this creates a batch job that generates all the DDL into a work statement list named DSG25D0P. We submit this batch job and **PF3** out and return to the database panel ADB21D.

On the command line, we issue the **WSL** command to get to the work statement list panel as shown in Figure 5-11.

```
ADB21D in ------------------- DB8A Databases ---------------------- Row 1 of 9
Command ===> wsl                                          Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                        Storage  Buffer          Created      Index
Select Name     Owner    Group    Pool       DBID By       T E Buffer Pool
       *        *        *        *             * *        * * *
------ -------- -------- -------- -------- ------ -------- - - -----------
       DSG24D0  PAOLOR2  GLWG01   BP15        288 PAOLOR2    E BP16
       DSG24D01 PAOLOR2  GLWG01   BP15        294 PAOLOR2    E BP16
*EN    DSG24D02 PAOLOR2  GLWG01   BP15        295 PAOLOR2    E BP16
       DSG24D03 PAOLOR2  GLWG01   BP15        292 PAOLOR2    E BP16
       DSG24D04 PAOLOR2  GLWG01   BP15        296 PAOLOR2    E BP16
       DSG24D1  PAOLOR2  GLWG01   BP15        289 PAOLOR2    E BP16
       DSG24D12 PAOLOR2  GLWG1    BP15        303 PAOLOR2    E BP16
       DSG24D2  PAOLOR2  GLWG01   BP15        293 PAOLOR2    E BP16
```

Figure 5-11 Issue the WSL command to get to the Work Statement List panel

We then generate a job to run this WSL (how to do this is described in 3.9.6, “Executing a work statement list” on page 104), and execute it to create the new database DSG25D0P. The result, shown in Figure 5-12, is produced using the **DS** (Display Structure) line command.

```
ADB21DS n --------------- DB8A Database Structures --------------- Row 1 of 75
Command ===>                                                   Scroll ===> PAGE

Line commands:  S - Show object  DSN - Data sets
                                                          PSID/
Sel Type   Object Name              Qualifier  DBID ISOBID   OBID Note
    *      *                        *             *      *      * *
--- ------ -----------------------> -------- ------ ------ ------ -------------
    D----- DSG25D0P----------------              302      0      0
     S       TS25001                DSG25D0P    302      2      1 Segmented
      T        GLWTACT              DSG25D0P    302      0      3
       UC        GLWPACT                          0      0      0 Primary key
       X         GLWXACT1           DSG25D0P    302      5      4
       V         WLACT              DSG25D0P      0      0      0
      T        GLWTJBS              DSG25D0P    302      0      6
       X         GLWXJBS1           DSG25D0P    302      8      7
       V         WLJOBSALARY        DSG25D0P      0      0      0
      T        GLWTLCN              DSG25D0P    302      0      9
       X         GLWXLCN1           DSG25D0P    302     11     10
       V         WLLOCN             DSG25D0P      0      0      0
      T        GLWTLNM              DSG25D0P    302      0     12
       X         GLWXLNM1           DSG25D0P    302     14     13
       V         WLLASTNAME         DSG25D0P      0      0      0
      T        GLWTPGW              DSG25D0P    302      0     15
```

Figure 5-12 Structure of new database DSG25D0P with changed table space name

Notice that if the GEN command is directed to a sequential file (rather than a WSL) then we could have used the new IMPORT function on the receiving end to feed the DDL into the CM process. However, this seemed to be not an overly useful approach to get objects created. Hence we have decided to not focus on the IMPORT function related to that specific purpose.

Tip: GEN is also available as a primary command to generate DDL for a list of objects on a given panel (for example, multiple databases) and all their dependencies.

5.3 Developing databases using generated DDL

In this section we discuss some considerations regarding DDL.

5.3.1 Methods of generating DDL

DDL can be generated using data modeling tools, data repository capabilities, DDL received from a vendor, and so on.

5.3.2 Creating the Master Development Database (D0)

In this book we use the IBM DB2 Workload Generator (GLW) for the creation of the initial development database, using its dynamically generated and executed DDL. The purpose of this tool is to create and drive workload on a DB2 for z/OS database. The database can later be manipulated and cloned using the DB2 Administration Tool and Object Compare.

5.4 Rolling out the initial design (V0)

In this section we describe functions of DB2 Admin and DB2 Compare that you can use to roll out an initial design to multiple environments, like other development systems, acceptance test, volume test, and one or more production environments.

When the development test environment has reached its maturity, it can now be cloned as personal test environments, unit test environments for developers, or any other environment you have in your shop.

Using DB2 Admin and DB2 Compare, there are basically four different ways to produce the DDL for those systems you want to roll out to. All of them have their merits and deficiencies, for example, one technique allows you to roll out (copy) the data as well, whereas others only focus on the easy creation of the DDL (see Figure 5-13).

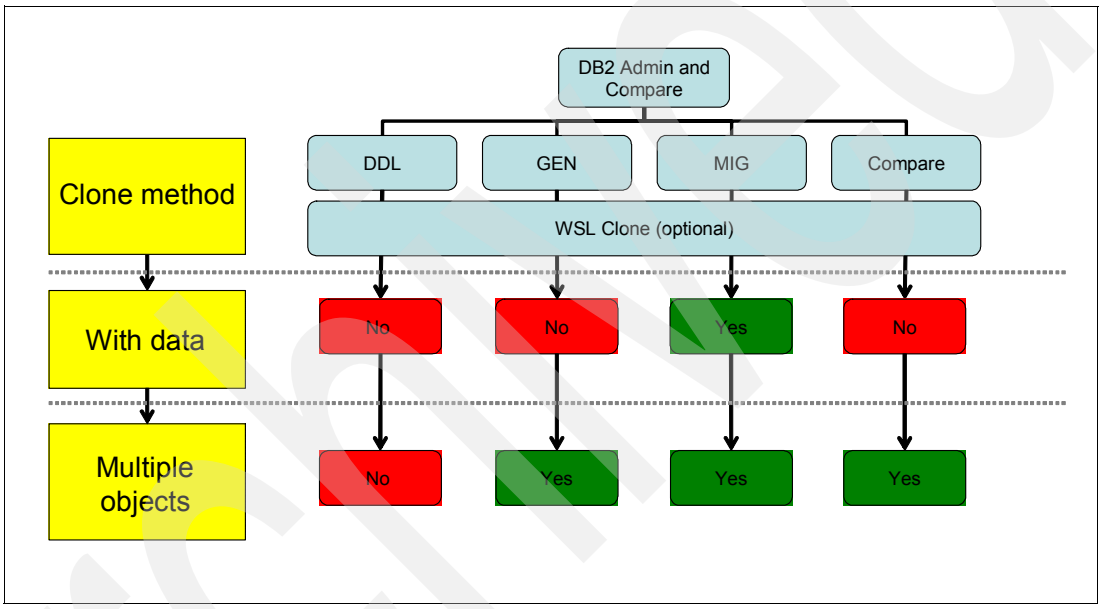

Figure 5-13 Showing the available options to clone DDL with data

1. GENerate command
2. MIGrate command
   a. DDL and data
   b. DDL only
   c. data only
3. Compare DDL or DB2 Catalog against a dummy file
4. WSL cloning

Important: With respect to the new Change Management functionality, there is little difference in the way an initial design is implemented and rolled out. However, the Compare option does show an example with the new CM functionality.

For executing the generated DDL on the target site, you could run it through DB2 Admin and register it as a change. This would be either via IMPORT or Option 2.1 (running from a data set).

5.4.1 Using generate (GEN) to clone a database

The generate (GEN) function of DB2 Admin enables you to generate the DDL for a database and all its “dependent” objects in one operation and also apply masking. However, using GEN, you cannot migrate the data as part of the generate operation. If that is a requirement, some of the other options, such as MIGrate, might be more suitable. Alternatively, for those tables that you must populate with data, you can use a manual unload-load process, or use the DB2 Test Database Generator for z/OS product.

Specifying **GEN** as a line command in front of a database, takes you to a panel where you can indicate which database objects and privileges you want to include in the generation. You can use full masking capabilities in GEN and can write the output to a work statement list if requested.

For a detailed description of GEN, see 5.2.2, “Creating many objects based on existing objects (GEN)” on page 204.

5.4.2 Creating (cloning) development databases (D1-D4) using MIG

To clone the master development database (D0) to each of the development environment clones (D1-D4), you can use the Migrate (MIG) function of DB2 Admin. Migrate is done in three steps and the clone is done within the same DB2, meaning that source and target environment are the same (see Figure 5-14).

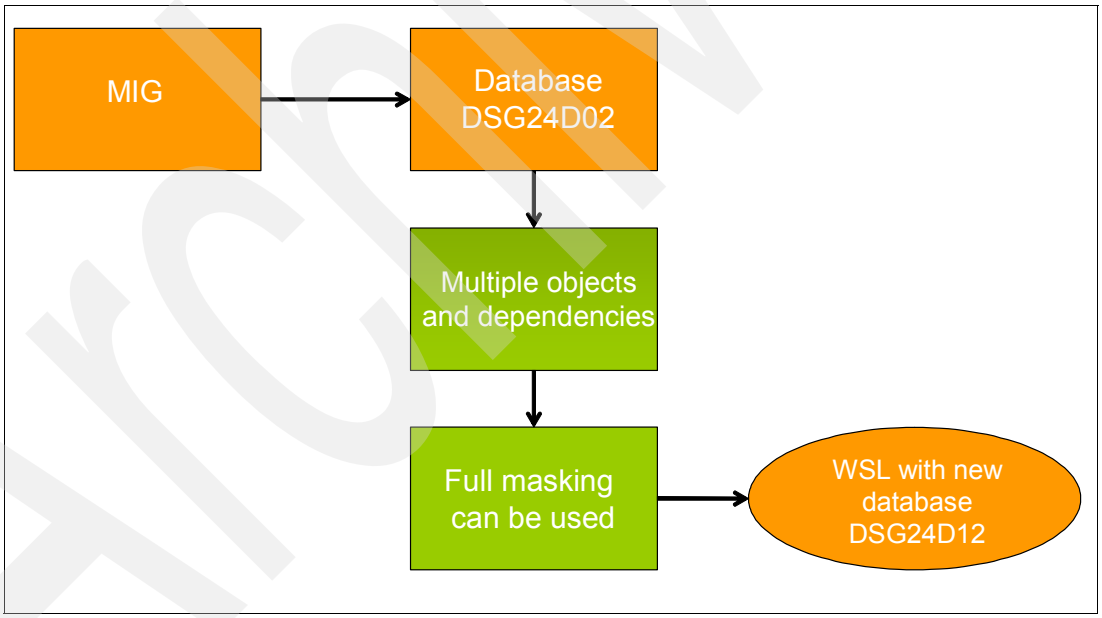

Figure 5-14 Clone with MIGRATE within the same DB2 subsystem

- Migrate panels to generate jobs
- Run jobs on the source system
- Run jobs on the target system

When using the MIGrate command, you can use the **ADD** command to add another database, or tables from another database, to be migrated at the same time.

MIGrate in non-WSL mode has the full masking functions that are available in GEN.

You can choose whether you want to migrate only the DDL, the data, or both.

Migrate generates jobs to perform the requested tasks. When using migrate, you can request to have the database on the target system dropped first, take image copies, and run various utilities like CHECK DATA and RUNSTATS.

When the target database is in the same subsystem as the source database, you can submit the jobs locally. The way to migrate to another subsystem is described in 5.4.3, “Cloning to another DB2 subsystem (MIGRATE)” on page 221.

If you are migrating the database to another subsystem, you have to make sure that you can access the DDL files, as well as unload files from the other subsystem, so that you are able to use them in the new subsystem.

Let us now look at the migration process in more detail.

Migration panels

In our sample database development environment, our target clones (D1 - D4) are on the same DB2 subsystem as the source master (D0). Therefore all the jobs can run on the same machine. We only change the owner IDs for all objects, as well as the database name. The data is also migrated.

Migration panels to set up the actual jobs

To select the DB2 Admin MIGrate function, we first select option **1** (DB2 System Catalog) from the DB2 Administration Menu. Then we enter the D0 database name (**DSG24D02**), select option **D** and press Enter (see Figure 5-15) to list the database.

```
DB2 Admin --------------------- DB8A System Catalog --------------------- 13:31
Option ===> D

                                                                More:     +
Object options:                                          DB2 System: DB8A
  AO - Authorization options                             DB2 SQL ID: PAOLOR2
   G - Storage groups                P - Plans
   D - Databases                     L - Collections
   S - Table spaces                  K - Packages
   T - Tables, views, and aliases    M - DBRMs
   V - Views                         H - Schemas
   A - Aliases                       E - User defined data types
   Y - Synonyms                      F - Functions
   X - Indexes                       O - Stored procedures
   C - Columns                       J - Triggers
   N - Constraints                   Q - Sequences
  DS - Database structures         DSP - DS with plans and packages

Enter standard selection criteria (Using a LIKE operator, criteria not saved):
Name     ===> DSG24D02           >  Grantor  ===>          >
Owner    ===>          >            Grantee  ===>          >
In D/L/H ===>                    >  Switch Catalog Copy  ===> N   (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
Column   ===>                    >  Operator ===>        Value ===>
```

Figure 5-15 System Catalog panel

This takes you to the Databases panel. Tab to the database you want to migrate and type **MIG** or you can enter **MIG** on the command line to migrate all objects in the list (see Figure 5-16).

Note that **MIG** is both a line command (Figure 5-16) and a primary command (Figure 5-17).

```
DB2 Admin ------------------- DB8A Databases ---------------------- Row 1 of 1
Command ===>                                                   Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                        Storage  Buffer          Created      Index
Select Name     Owner    Group    Pool       DBID By       T E Buffer Pool
       *        *        *        *            * *        * * *
------ -------- -------- -------- -------- ------ -------- - - -----------
MIG    DSG24D02 PAOLOR2  GLWG01   BP15        295 PAOLOR2    E BP16
****************************** END OF DB2 DATA ******************************
```

Figure 5-16 Selecting Migration source database

After pressing Enter, the Migrate Table Spaces panel (see Figure 5-17) appears. This panel lists all the table spaces to be migrated. You can add objects to the list by using the **ADD** primary command. If you want to make sure that you migrate complete referential structures, type **ADDRI** (as a line command) in front of an object in the list. This adds RI related objects to the list. Type **MIGRI** to add all related table spaces to the migration operation. If you are satisfied with the list of objects you want to migrate, enter **MIG** on the command line.

```
 DB2 Admin ------------------- DB8A Migrate Table Spaces ---------- Row 1 of 14
Command ===> mig                                               Scroll ===> PAGE
Commands: MIG - Generate jobs  ADD - Add databases
          MIGRI - Add all RI related table spaces
Line commands:
 D - Delete  T - Show tables  ADDRI - Add RI related table spaces
 RIT - Show RI related tables

       Data     Table           No of           Table RI           VSAM
Select Base     Space      Part Tables LOB   Relations Added     KB Used
       *        *             *      * *             * *               *
------ -------- -------- ------ ------ --- ----------- ----- -----------
       DSG24D02 GLWSDPT       0      1 NO            0 NA           1440
       DSG24D02 GLWSEMP       1      1 NO            0 NA            144
       DSG24D02 GLWSEMP       2      1 NO            0 NA            144
       DSG24D02 GLWSEMP       3      1 NO            0 NA            144
       DSG24D02 GLWSEMP       4      1 NO            0 NA            144
       DSG24D02 GLWSEPA       0      1 NO            0 NA           1440
       DSG24D02 GLWSPJA       0      1 NO            0 NA           1440
       DSG24D02 GLWSPRJ       0      1 NO            0 NA           1440
       DSG24D02 GLWSSPL       1      1 NO            0 NA             96
       DSG24D02 GLWSSPL       2      1 NO            0 NA             96
       DSG24D02 GLWSSPL       3      1 NO            0 NA           1440
```

Figure 5-17 Selecting migration for all table spaces

Pressing Enter now brings you to the Migrate Parameters panel (see Figure 5-18).

```
ADB28M in ------------------- DB8A Migrate Parameters ------------------- 18:56
Option ===>

Please specify the following for DB2 Admin Migrate:      DB2 System: DB8A
                                                         DB2 SQL ID: PAOLOR2
                                                                More:     +
Worklist name . . . . . . . : MIGDATA  (also used as middle qualifier in DSNs)

Data set information:
  PDS for batch jobs  . . . : 'ADMIN.MIG.JCL'
  Prefix for datasets . . . : PAOLOR2

Target system parameters:
  DB2 subsystem id (SSID) . : DB8A        DB2 release  . . . . :
  Target system node name . :             Submit job at local. : NO  (Yes/No)
  DB2 load library  . . . . :
  DB2 sample pgm load lib . :
  New TS storage group  . . : GLWG01   >  New IX storage group : GLWG01   >
  New database  . . . . . . : DSG24D12    New owner of objects : DSG24D12 >
  New grantor . . . . . . . :          >
  Catalog qualifier . . . . :          >  (default SYSIBM , only applicable
                                          when scope contains catalog stats.)
```

Figure 5-18 Migrate Parameters panel 1/3

During our migration, we specify the following values on the Migrate Parameters panel:

Worklist name	We choose **MIGDATA** for worklist name since we are migrating D0 to D1 with data.
PDS for batch jobs	We choose **ADMIN.MIG.JCL** as PDS name for the migrate batch jobs, since we have different sets of files to better separate our jobs.
Prefix for data sets	We choose **PAOLOR2** as the prefix for our data sets. It is the logon ID of the person creating the jobs. In a real-life environment, you might want to use a prefix that contains the database name or the project name and include some type of date/time qualifier as well. This way you avoid a situation in which existing jobs are overwritten by accident.
DB2 subsystem ID (SSID)	We are cloning to the same subsystem as the master, which is **DB8A**.
DB2 release	We left this blank since the target is the same DB2 as the source.
Target system node name	Since we are targeting the same subsystem as the master, we leave this blank.
Submit jobs at local	We choose **N**, since entering Y requires us to fill in the Target system node name, even though we are going to the same subsystem.
Load libraries	We left this blank because we work with work statement lists and because we stay on the same target as the source.
New Storage group	We choose **GLWG01** for both table space and index storage groups which is the same storage group used by the master database. If you are going to a different DB2 subsystem, you would probably want to change to a storage group used by that subsystem.

New database	Since we are cloning D0 to D1, we choose **DSG24D12** as the target database.
New owner of objects	Since we are cloning D0 to D1, we choose **DSG24D12** as the target object owner. We suggest that you specify an owner name more generic across databases.

Note: In DB2 Admin Version V7.2 and when working with WSL, the only name changes you can perform during MIGrate are changing the database name, the table space name, the index space name, the storage group name, and the object owner name.

If WSL is not used then full masking is supported:
`Generate work stmt list . . . . . : No  (Yes/No)`

Pressing **PF8** allows you to specify more options (see Figure 5-19).

```
ADB28M in ------------------- DB8A Migrate Parameters ------------------- 19:02
Option ===>

Please specify the following for DB2 Admin Migrate:      DB2 System: DB8A
                                                         DB2 SQL ID: PAOLOR2
                                                                More:   - +
Migrate options:
  Generate work stmt list . . . . . : YES      (Yes/No)
  Use masking for batch migrate . . : NO       (Yes/No, N if stmt list is Y)
  Combine job steps . . . . . . . . : NO       (Yes/No)
    Member prefix for combined jobs : MIGDA    (default ADBMG )
  Scope of migrate:
    DDL . . . . . . . . . . . . . . : YES      (Yes/No)
    Data  . . . . . . . . . . . . . : YES      (Yes/No)
    Catalog statistics  . . . . . . : NO       (Yes/No)
  DROP on target before CREATE  . . : NO       (Yes/No,No if scp. is data only)
  Create storage group  . . . . . . : NO       (Yes/No)
  Run SQLID . . . . . . . . . . . . : <NONE>   (Blank, a SQLID, or <NONE>)
  Unload method . . . . . . . . . . : U        (U - Unload)
  Parallel utilities  . . . . . . . : YES      (Yes/No)

Optional steps after reload:
  Run CHECK DATA  . . . . . . . . . : YES      (Yes/No)
  Run RUNSTATS  . . . . . . . . . . : YES      (Yes/No)
  Run IMAGE COPY  . . . . . . . . . : NO       (Yes/No)
  Run REBIND  . . . . . . . . . . . : NO       (Yes/No)
```

Figure 5-19 Migrate Parameters panel 2/3

Generate work stmt list	We choose **Y**, to add the generated jobs in a worklist.
Use masking for batch migrate	We must choose **N** because masking is not allowed with WSLs on the MIGRATE.
Combine job steps	We choose **N** for Combine job steps. For migrate with WSL, this option does not matter.
Member prefix for combined job steps	We choose **MIGDA** to indicate migrate with data.
Scope of migrate with DDL	We choose **YES** for DDL.
Scope of migrate with Data	We choose **YES** for data.

Scope of migrate with catalog statistics	We choose NO to not migrate the catalog statistics because we intend to specify that RUNSTATS is to be run on the target environment.
DROP on target before CREATE	We choose **N** for DROP on target before CREATE because we are creating a brand new database and there are no objects to drop.
Create Storage Group	We choose **N** for Create Storage Group since we are using the same storage group as D0.
Run SQLID	We use the job submitters userid (PAOLOR2) as SQLID to generate the “SET CURRENT SQLID.

Attention: Ensure that CREATEDBA, USE OF STOGROUP, and USE OF BUFFERPOOL privileges have been granted to this SQLID before running the migrate jobs.

Unload Method	This can be IBM UNLOAD utility or the High Performance Unload (HPU) utility. We choose **U** for IBM UNLOAD.
Parallel Utilities	Run IBM DB2 Utilities in parallel. We choose **N.**
Run CHECK DATA	We choose **Y** for the Run CHECK DATA option since we are migrating data and want to ensure the data conforms to the RI constraints.
Run RUNSTATS	We choose **Y** for the Run RUNSTATS option since we do not migrate the catalog statistics and want to ensure that statistics are current after the object gets migrated.
Run IMAGE COPY	We choose **Y** for the Run IMAGE COPY option since we want to take a full copy the objects after the migration.
Run REBIND	We choose **N** for the Run REBIND option since we do not have plans or packages.

When you press PF8 to scroll down the panel, you see additional options on the panel as shown in Figure 5-20.

```
Utility control options:
 Generate template statements . . . : NO       (Yes/No)

BP - Change batch job parameters
TU - Specify template usage
```

Figure 5-20 Migrate Parameters panel 3/3

Generate template statements	We choose **NO** for this option.

After entering all the above fields on the Migrate Parameters panel, we press Enter, which then brings us to the panel for specifying work statement list information (see Figure 5-21).

```
EsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
e DB2 Admin ------------------ Specify Work Statement List ----------------- e
e                                                                            e
e                                                                            e
e  Work stmt list dsn  . . . 'ADMIN.MIG.WSL'                                 e
e  Work stmt list name . . . MIGDATA                                         e
e                                                                            e
e                                                                            e
e                                                                            e
e                                                                            e
e                                                                            e
e                                                                            e
e                                                                            e
e                                                                            e
DssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

Figure 5-21 Specify Work Statement List panel

After pressing Enter again, this opens a pop-up panel showing the progress of job generation. This panel is shown (see Figure 5-22) because it might take some time to build the jobs, so that you do not get concerned about the waiting time.

```
EsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
e DB2 Admin ----------- Migrate Progress ---------------- e
e                                                        e
e The migrate JCL jobs are being created.  This process   e
e may take several minutes.                              e
e                                                        e
e Status :                                               e
e Building Convert Load Card JCL                         e
e                                                        e
e Hit ATTN if you want to interrupt it and then          e
e type HI to terminate it.                               e
e                                                        e
e                                                        e
DssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

Figure 5-22 Migrate Progress panel

When the job generation is finished, it places the jobs in the PDS, **ADMIN.MIG.JCL**, and opens an ISPF EDIT session with that library selected (see Figure 5-23).

```
EDIT     ADMIN.MIG.JCL                                        Row 00001 of 00006
Command ===>                                                    Scroll ===> PAGE
   Name     Prompt          Size    Created           Changed            ID
. MIGCOSE                   101   2007/03/01   2007/03/01 20:52:07    PAOLOR2
. MIGCOS1                   818   2007/03/01   2007/03/01 20:52:07    PAOLOR2
. MIGDASE                   101   2007/03/02   2007/03/02 18:34:11    PAOLOR2
. MIGDAS1                   820   2007/03/02   2007/03/02 18:34:10    PAOLOR2
. MIG2SE                    101   2007/03/02   2007/03/02 18:27:40    PAOLOR2
. MIG2S1                    820   2007/03/02   2007/03/02 18:27:40    PAOLOR2
  **End**
```

Figure 5-23 Migrate Jobs

Archived

Because we have specified to generate a work statement list, we actually get two jobs in our scenario:

MIGDASE Must be run as last job on the source environment
MIGDAS1 Must be run as first job on the source environment. The unload of data is one step per table space and the unload of data is not in work statement list format. A work statement list is generated to build the objects at the target and load the data.

If there are more than 254 steps (JCL limit) to unload the data in one job, then there are more job members:

MIGDAS2 Must be run second (if there are more than 254 steps in MIGDAS1
MIGDAS3 Must be run third
MIGDASn Must be run one after the other

If no data is migrated, there are always two members (xxxxS1 and xxxxxSE). It is now time to execute the jobs in the right order. After S1 is finished, we have the unloaded files and the generated work statement list (see Figure 5-24).

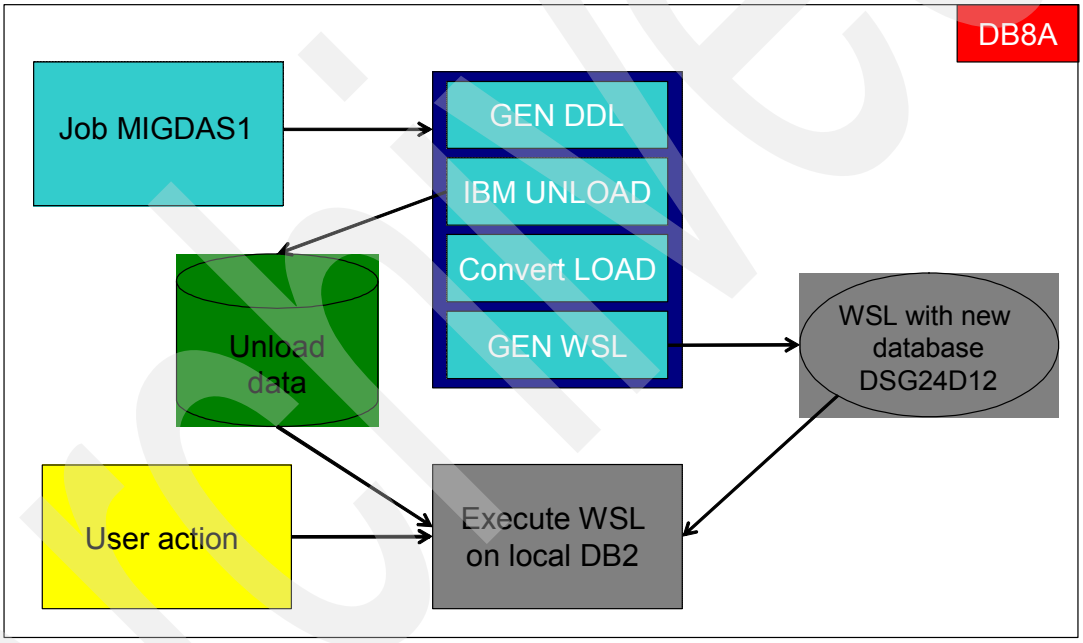

Figure 5-24 Job flow for cloning a database within same DB2 subsystem

Sequence of events on the source:

1. Submit Job MIGDAS1:
   a. Re-engineer the DDL.
   b. Unload data.
   c. Convert load control files.
   d. Generate the WSL file.
2. User action (only required, if target cannot access data on the source system):
   – Copy unload files and WSL to the target system
3. Submit Job MIGDASE (only required for housekeeping purposes)
   – Delete source system data sets

Attention:

Do not run the xxxxxSE job (in our case MIGDASE) until you have transferred the unload files and work statement list to the target environment. The xxxxxSE job cleans up and deletes the unload data sets. Otherwise you have to resubmit the xxxxxS1 - xxxxxSn jobs to recreate these files again.

Do not run this job if you do not want to transfer the data — it would delete your unloaded data and you would have to recreate the unload file and the WSL.

Sequence of events on the target:

1. User action (only required, if target cannot access data on source)
   - Transfer of unload data sets and work statement list to target is complete.
2. User action:
   a. Generate JCL to run the WSL. The WSL does the following tasks:
      - CREATE objects
      - LOAD
      - CHECK
      - RUNSTATS
      - COPY
   b. Execute the JCL to run the WSL.

When the WSL finishes, we have cloned our source database DSG24D02 so that target database DSG24D12 has been successfully created with all dependent objects and with data.

5.4.3 Cloning to another DB2 subsystem (MIGRATE)

In our development scenario (see Part 3, “Change Management scenario” on page 199). We described the acceptance tests environment on another DB2 subsystem. Our acceptance test system is the DB2 subsystem **D8F1**.

To migrate a DB2 object to a remote system, we use in general the same procedure as when migrating to a local system (as described in 5.4.2, “Creating (cloning) development databases (D1-D4) using MIG” on page 213). Since we use the work statement list (WSL) technique, there is not much difference in the way this works.

The first set of jobs is run on the source system. Then the created output has to be transferred to the remote system, unless you are using shared DASD. When the files are shared, the jobs to create and load the objects on the target site can be submitted.

Figure 5-25 shows a diagram of the cloning scenario.

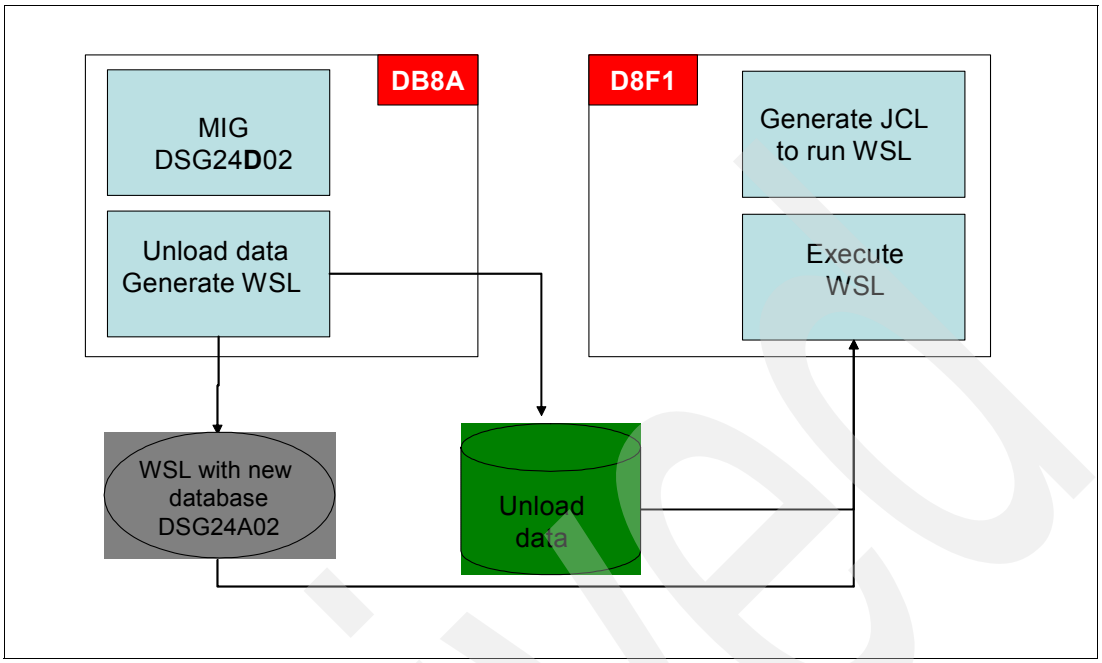

Figure 5-25 Clone into another DB2 subsystem

Migration panels

In our sample database environment, the acceptance master (A0) is on another DB2 subsystem as the source master (D0). We only change the database name and the owner IDs for all objects. We also migrate all data. In addition we have to specify the target system.

Migration panels to set up jobs

Selecting the objects for the migration is done in the same way as shown in Figure 5-15 on page 214 through Figure 5-17 on page 215.

After that, the Migrate Parameters panel appears, as shown in Figure 5-26.

```
DB2 Admin ------------------- DB8A Migrate Parameters ------------------- 11:37
Option ===>

Please specify the following for DB2 Admin Migrate:      DB2 System: DB8A
                                                         DB2 SQL ID: PAOLOR2
                                                                   More:     +
Worklist name . . . . . . . : MIGD8F1  (also used as middle qualifier in DSNs)

Data set information:
  PDS for batch jobs  . . . : 'ADMIN.MIG.JCL'
  Prefix for datasets . . . : PAOLOR2

Target system parameters:
  DB2 subsystem id (SSID) . : D8F1        DB2 release  . . . . :
  Target system node name . :             Submit job at local. : NO  (Yes/No)
  DB2 load library  . . . . :
  DB2 sample pgm load lib . :
  New TS storage group  . . : GLWG01   >  New IX storage group : GLWG01   >
  New database  . . . . . . : DSG24A02    New owner of objects : DSG24A02 >
  New grantor . . . . . . . :          >
  Catalog qualifier . . . . :          >  (default SYSIBM , only applicable
                                          when scope contains catalog stats.)
```

Figure 5-26 Migrate Parameters panel for remote target system 1/2

On the Migrate Parameters panel, we enter the following values:

Worklist name	We choose **MIGD8F1** for worklist name since we are cloning to DB2 subsystem D8F1.
PDS for batch jobs	We choose **ADMIN.MIG.JCL** as the name of the PDS to store all generated migrate batch jobs.
Prefix for data sets	We choose **PAOLOR2** as the prefix for data sets. This is the logon ID that is creating the jobs.
DB2 subsystem id (SSID)	We are cloning to another subsystem as the master. The remote subsystem is called **D8F1**.
Target system node name	This parameter is not used when working with work statement lists. So we leave it blank.
Submit jobs at local	This parameter is not used when working with work statement lists. We choose **N.**
Load libraries	We do not have to specify the target DB2 load library and DB2 sample program library. Because we use a work statement list on the target there is no requirement for it. So we leave this blank.
New Storage group	We choose a new storage group **GLWG01** for the target system for the table spaces as well for the indexes. We choose the same name as we used for the source system.
New database	We are cloning D0 to A0. Therefore, we choose **DSG24A0G** as the target database.
New owner of objects	Since we are cloning D0 to A0, we use **DSG24A02** as the target object owner name. Make sure you set up a good naming standard for object owners to help you to define the owner for different databases in each environment.

Archived

Pressing PF8 now brings us to the second part of the Migrate Parameters panel, where additional parameters must be specified (see Figure 5-27).

```
DB2 Admin ------------------- DB8A Migrate Parameters ------------------- 11:37
Option ===>

Please specify the following for DB2 Admin Migrate:      DB2 System: DB8A
                                                         DB2 SQL ID: PAOLOR2
                                                                   More:   - +
Migrate options:
  Generate work stmt list . . . . . : YES      (Yes/No)
  Use masking for batch migrate . . : NO       (Yes/No, N if stmt list is Y)
  Combine job steps . . . . . . . . : NO       (Yes/No)
    Member prefix for combined jobs : MD8F1    (default ADBMG )
  Scope of migrate:
    DDL . . . . . . . . . . . . . . : YES      (Yes/No)
    Data  . . . . . . . . . . . . . : YES      (Yes/No)
    Catalog statistics  . . . . . . : NO       (Yes/No)
  DROP on target before CREATE  . . : NO       (Yes/No,No if scp. is data only)
  Create storage group  . . . . . . : NO       (Yes/No)
  Run SQLID . . . . . . . . . . . . : <NONE>   (Blank, a SQLID, or <NONE>)
  Unload method . . . . . . . . . . : U        (U - Unload)
  Parallel utilities  . . . . . . . : NO       (Yes/No)

Optional steps after reload:
  Run CHECK DATA  . . . . . . . . . : YES      (Yes/No)
  Run RUNSTATS  . . . . . . . . . . : YES      (Yes/No)
  Run IMAGE COPY  . . . . . . . . . : NO       (Yes/No)
  Run REBIND  . . . . . . . . . . . : NO       (Yes/No)
```

Figure 5-27 Migrate Parameters panel for remote target system 2/2

Generate work stmt list	We choose **Y**, to add the generated jobs in a worklist. However, the unload of data is still a batch job. What gets generated into the WSL is the CREATE of all the objects at the target site and the loading of the data and the utilities selected for it (CHECK and RUNSTATS, in our case).
Use masking for batch migrate	We must choose **N** because masking is not allowed with WSLs on the MIGRATE.
Combine job steps	We choose **N** because this option applies only if batch JCL is generated. For migrate with WSL, this option does not matter.
Member prefix for combined job steps	We choose **MD8F1** as member name prefix. Although we are not combining job steps, this option sets the member prefix for the jobs in the generated work statement lists
Scope of migrate	We choose **B** to migrate both the database structure and the data.
DROP on target before CREATE	We choose **N** for DROP on target before CREATE because we are creating a brand new database and there are no objects to drop.

Archived

Create Storage Group	We choose **N** for Create Storage Group because we assume that the same storage group as for D0 already exists on the target. There are other applications on the D8F1 subsystem.
Run SQLID	Following our naming conventions, we use **DSG24A02** as the CURRENT SQLID when the DDL is executed.

Attention: Ensure that CREATEDBA, USE STOGROUP, and USE BUFFERPOOL privileges have been granted to this SQLID before running the migrate jobs.

Unload Method	Can be IBM UNLOAD utility or the High Performance Unload (HPU) utility. We choose **U** for IBM UNLOAD.
Parallel Utilities	Run IBM DB2 Utilities in parallel. We choose **N.**
Run CHECK DATA	We choose **Y** for Run CHECK DATA, since we are migrating data and want to ensure that the data conforms to the RI constraints.
Run RUNSTATS	We choose **Y** for Run RUNSTATS, since we want to update the DB2 system catalog with the statistics.
Take an image copy	We choose **N** for Take an image copy, since we want to keep the load files.
Run REBIND	We choose **N** for the Run REBIND option, since this is a brand new database and there are no existing plans of packages.

And then there are more options on that panel. If you PF8 down, you can see the additional options as shown in Figure 5-28.

```
Utility control options:
 Generate template statements . . . : NO       (Yes/No)

BP - Change batch job parameters
TU - Specify template usage
```

Figure 5-28 Migrate Parameters panel 3/3

Generate template statements	We choose **NO** for this option.

After entering all the above fields on the Migrate Parameters panel, we press Enter, which then brings us to the panel for specifying work statement list information (see Figure 5-29).

Archived

```
Esssssss
```

As we requested to generate WSL, you see only two jobs in the PDS. They all start with MD8F1 as requested in the *Member prefix for combined jobs option* in Figure 5-27.

MD8F1SE This job is a housekeeping job that deletes the data sets after the migration is done. It also tells you which data sets have to be transmitted to the target system.

MD8F1S1 This is the job that runs on the source system. It does the unload of the data and generates the work statement list that is later to be run on the target system.

So we can conclude that there was not much difference in the way this migrate was set up and executed for a remote DB2. When you use WSLs, the only actions on the source that are required are to unload the data and generate the WSL to be executed on the target. Then these files (unload files and WSL) must get transferred to the target system. And on the target DB2, Admin picks up the WSL, which does all the required work (CREATE, LOAD, CHECK, RUNSTATS in our case). This work flow is much the same as with local migrates. Only the panel specifications on the local site must be adjusted for the remote DB2 subsystem name.

5.4.4 Cloning multiple physical databases (WSL cloning and COMPARE)

It is not unusual to have multiple physical databases (DB2 databases) for a single logical database (usually an application or subset). This is normally done to reduce the size and contention on the database descriptor (DBD). Some shops have a standard of one physical database per table space. This section describes two techniques to create a clone of a logical database with multiple physical databases as shown in Figure 5-32.

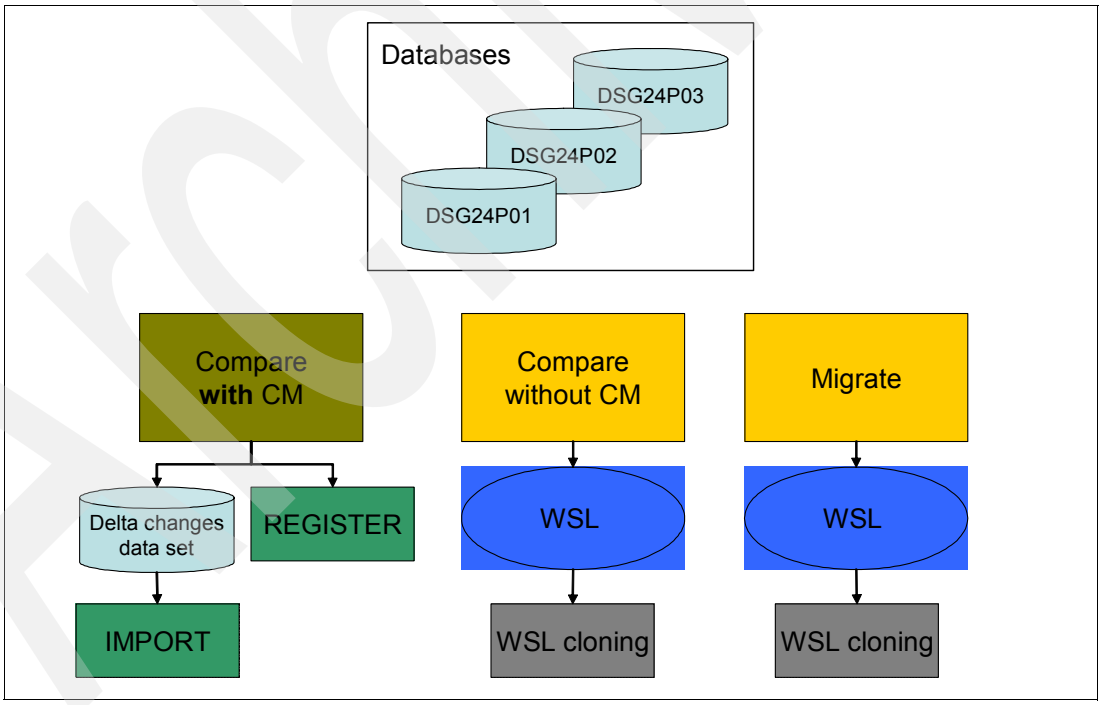

Figure 5-32 Migrate and Compare technique to clone multiple databases

- DB2 Compare:
  - Does not allow you to migrate data
  - If working with new CM function: DELTA file for later IMPORT or ANALYZE
  - If working without CM function: WSL file for later WSL cloning

- MIGRATE:
  - Does allow you to migrate data
  - Does not work with new CM function:
    - MIG as primary command
    - ADD as primary command
  - Results in WSL file for later WSL cloning

Which technique you want to use depends on the number of environments you have to clone and whether or not you have to migrate data as well.

- DB2 Compare:
  - If using the new CM functionality: this writes the work that has to be done to migrate the (multiple) databases to a DELTA changes file or you can register the work on the local DB2.
    - If staying on the local subsystem and having the changes registered with COMPARE you later ANALYZE and RUN the clone.
    - If going to another DB2 subsystem you transfer the DELTA file (coming out of the COMPARE process) and IMPORT it on the other DB2 to get the changes registered. This then can be followed by ANALYZE and RUN to get the clone done.
  - If not using the new CM functionality: this can write the work that has to be done to migrate the (multiple) databases to a work statement list. Once you have generated one WSL, you can easily clone it to be used for any other environment that requires the set of databases.
  - However, using the COMPARE technique you cannot migrate any data.
- MIGRATE:
  - When using the MIGrate function, all the operations have to be repeated for every environment you want to migrate to. When the object names differ between environments they can be changed with WSL cloning and with masking for batch migrate if non-WSL option is chosen. However, data can be migrated as well using this technique.

We assume we have an existing logical master *production* database **DSG24P0** with one physical database **DSG24P0n** for each table space. Now we want to clone these physical **DSG24P0n** databases, to physical databases **DSG24P1n**.

Using MIGRATE - MIG primary command

First we get a list of databases we want to have migrated. In our case we could search on **DSG24P*** and get all databases listed which qualify for this. We then can process them all on one pass by typing **mig** in the command line — this adds all databases that are on this list to the migrate process (as shown in Figure 5-33). We then press Enter.

```
DB2 Admin ------------------- DB8A Databases ---------------------- Row 1 of 3
Command ===> mig                                            Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                         Storage  Buffer          Created      Index
Select Name     Owner    Group    Pool       DBID By       T E Buffer Pool
       *        *        *        *             * *        * * *
------ -------- -------- -------- -------- ------ -------- - - -----------
       DSG24P01 PAOLOR2  GLWG01   BP15        308 PAOLOR2    E BP16
       DSG24P02 PAOLOR2  GLWG01   BP15        309 PAOLOR2    E BP16
       DSG24P03 PAOLOR2  GLWG01   BP15        310 PAOLOR2    E BP16
****************************** END OF DB2 DATA ******************************
```

Figure 5-33 Selecting migration source databases

If your database naming standards are not like this example, then it might get more difficult to have them appear on a list. If this is the case, you can migrate database by database with the **ADD** function.

Using MIGRATE - ADD function

In this case you start with **mig** as line command against a single database (as shown in Figure 5-34) — and then press Enter.

```
DB2 Admin ------------------- DB8A Databases ---------------------- Row 1 of 3
Command ===>                                                Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                         Storage  Buffer          Created      Index
Select Name     Owner    Group    Pool       DBID By       T E Buffer Pool
       *        *        *        *             * *        * * *
------ -------- -------- -------- -------- ------ -------- - - -----------
mig    DSG24P01 PAOLOR2  GLWG01   BP15        308 PAOLOR2    E BP16
       DSG24P02 PAOLOR2  GLWG01   BP15        309 PAOLOR2    E BP16
       DSG24P03 PAOLOR2  GLWG01   BP15        310 PAOLOR2    E BP16
****************************** END OF DB2 DATA ******************************
```

Figure 5-34 Migrate - Add function

This takes you to the panel, shown in Figure 5-35. There we use the **ADD** primary command to indicate that we want to migrate additional objects and press **Enter**.

```
 DB2 Admin ------------------- DB8A Migrate Table Spaces ---------- Row 1 of 14
Command ===> add                                            Scroll ===> PAGE

Commands: MIG - Generate jobs  ADD - Add databases
          MIGRI - Add all RI related table spaces
Line commands:
 D - Delete  T - Show tables  ADDRI - Add RI related table spaces
 RIT - Show RI related tables

       Data     Table            No of           Table RI           VSAM
Select Base     Space      Part Tables LOB   Relations Added     KB Used
       *        *             *      * *             * *             *
------ -------- -------- ------ ------ --- ----------- ----- -----------
       DSG24P01 GLWSDPT       0      1 NO            0 NA          1440
       DSG24P01 GLWSEMP       1      1 NO            0 NA           144
       DSG24P01 GLWSEMP       2      1 NO            0 NA           144
       DSG24P01 GLWSEMP       3      1 NO            0 NA           144
       DSG24P01 GLWSEMP       4      1 NO            0 NA           144
       DSG24P01 GLWSEPA       0      1 NO            0 NA          1440
       DSG24P01 GLWSPJA       0      1 NO            0 NA           144
       DSG24P01 GLWSPRJ       0      1 NO            0 NA           144
       DSG24P01 GLWSSPL       1      1 NO            0 NA            96
       DSG24P01 GLWSSPL       2      1 NO            0 NA            96
```

Figure 5-35 Add more objects to this migrate operation

You can now specify the prefix of a database name that you want to add to the list of objects to migrate (see Figure 5-36). We enter **DSG24P** and press Enter.

```
 DB2 Admin ----------------- DB8A Migrate Add Database ------------------ 14:25
Option ===>

Enter the partial name of the data base you want to add to the migrate
operation:

  Partial data base name . . : DSG24P

Press Enter to search for the data base.
```

Figure 5-36 Migrate Add Database panel

Select all databases you want to add by using the **S** line command and press Enter again (see Figure 5-37). Note that we do *not* select **DSG24P01**, because that database is already selected by the **MIG** command.

```
DB2 Admin ------------------- DB8A Migrate Add Database ----------- Row 1 of 4
Command ===>                                                  Scroll ===> PAGE

Line commands:
 S - Select

       Data
Select Base
       *
------ --------
       DSG24P0
       DSG24P01
S      DSG24P02
S      DSG24P03
****************************** END OF DB2 DATA ******************************
```

Figure 5-37 Selecting objects to be added to the migration list

After the objects have been processed press PF3 to return to the Migrate Table Spaces panel shown in Figure 5-38.

```
 DB2 Admin ------------------- DB8A Migrate Table Spaces --------- Row 12 of 42
Command ===> mig                                              Scroll ===> PAGE

Commands: MIG - Generate jobs  ADD - Add databases
          MIGRI - Add all RI related table spaces
Line commands:
 D - Delete  T - Show tables  ADDRI - Add RI related table spaces
 RIT - Show RI related tables

       Data     Table            No of           Table RI           VSAM
Select Base     Space      Part Tables LOB   Relations Added     KB Used
       *        *             *      * *            * *               *
------ -------- -------- ------ ------ --- ----------- ----- -----------
       DSG24P01 GLWSSPL       4      1 NO            0 NA             96
       DSG24P01 GLWS001       0      7 NO            0 NA            240
       DSG24P01 GLWS002       0      4 NO            0 NA           1440
       DSG24P02 GLWSDPT       0      1 NO            0 NA           1440
       DSG24P02 GLWSEMP       1      1 NO            0 NA            144
       DSG24P02 GLWSEMP       2      1 NO            0 NA            144
       DSG24P02 GLWSEMP       3      1 NO            0 NA            144
       DSG24P02 GLWSEMP       4      1 NO            0 NA            144
       DSG24P02 GLWSEPA       0      1 NO            0 NA           1440
       DSG24P02 GLWSPJA       0      1 NO            0 NA            144
```

Figure 5-38 Additional objects in the migration list

Both databases (**DSG24P02** and **DSG24P03**) have been added to the list of objects to be migrated. In this example, each database contains multiple table spaces.

So now that we have selected our multiple databases to be cloned, we follow the scenario as shown in Figure 5-39.

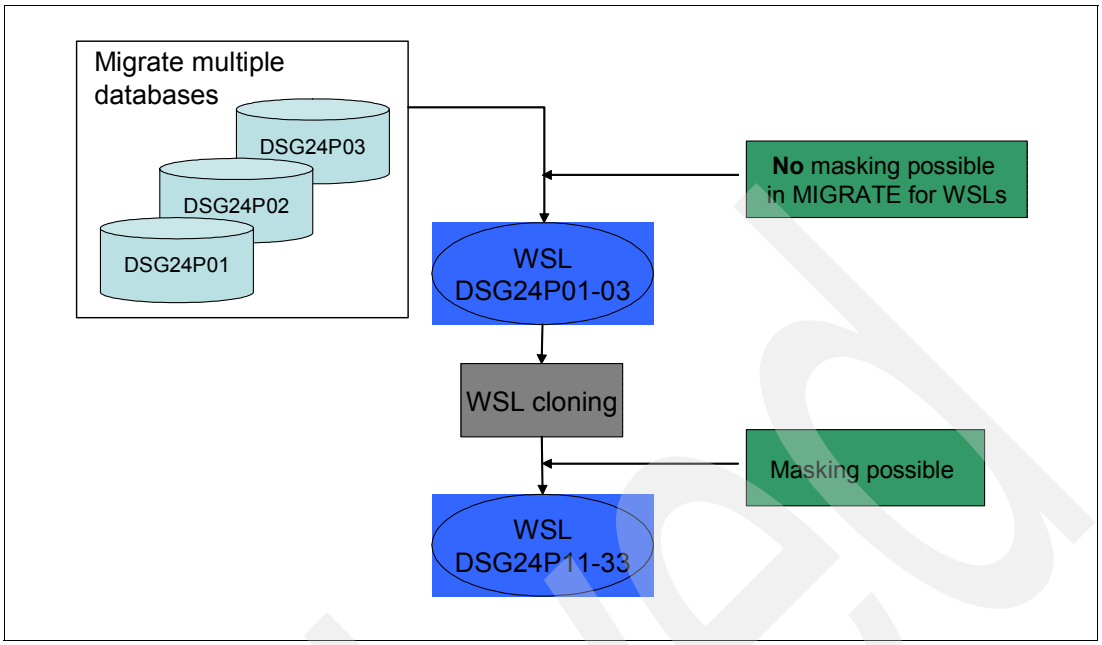

Figure 5-39 Scenario for cloning multiple databases with MIGRATE

If all objects you want to migrate are now displayed in the *updated* Migrated Table Spaces panel, type **MIG** and press Enter to indicate that you are ready to generate the jobs to migrate the objects. This takes you to the Migrate Parameters panel, shown in Figure 5-40.

Archived

```
DB2 Admin ------------------- DB8A Migrate Parameters ------------------- 14:35
Option ===>

Please specify the following for DB2 Admin Migrate:      DB2 System: DB8A
                                                         DB2 SQL ID: PAOLOR2
                                                                   More:     +
Worklist name . . . . . . . : MADD     (also used as middle qualifier in DSNs)

Data set information:
  PDS for batch jobs  . . . : 'ADMIN.MIG.JCL'
  Prefix for datasets . . . : PAOLOR2

Target system parameters:
  DB2 subsystem id (SSID) . : DB8A        DB2 release  . . . . :
  Target system node name . :             Submit job at local. : NO  (Yes/No)
  DB2 load library  . . . . :
  DB2 sample pgm load lib . :
  New TS storage group  . . :          >  New IX storage group :          >
  New database  . . . . . . :             New owner of objects :          >
  New grantor . . . . . . . :          >
  Catalog qualifier . . . . :          >  (default SYSIBM , only applicable
                                          when scope contains catalog stats.)
Migrate options:
  Generate work stmt list . . . . . : YES      (Yes/No)
  Use masking for batch migrate . . : NO       (Yes/No, N if stmt list is Y)
  Combine job steps . . . . . . . . : NO       (Yes/No)
    Member prefix for combined jobs : MADD     (default ADBMG )
  Scope of migrate:
    DDL . . . . . . . . . . . . . . : YES      (Yes/No)
    Data  . . . . . . . . . . . . . : YES      (Yes/No)
    Catalog statistics  . . . . . . : NO       (Yes/No)
  DROP on target before CREATE  . . : NO       (Yes/No,No if scp. is data only)
  Create storage group  . . . . . . : NO       (Yes/No)
  Run SQLID . . . . . . . . . . . . : <NONE>   (Blank, a SQLID, or <NONE>)
  Unload method . . . . . . . . . . : U        (U - Unload)
  Parallel utilities  . . . . . . . : NO       (Yes/No)
```

Figure 5-40 Migrate Parameters panel (1/2)

When we press PF8, we get additional options as shown in Figure 5-41.

```
Optional steps after reload:
  Run CHECK DATA  . . . . . . . . . : YES      (Yes/No)
  Run RUNSTATS  . . . . . . . . . . : YES      (Yes/No)
  Run IMAGE COPY  . . . . . . . . . : YES      (Yes/No)
  Run REBIND  . . . . . . . . . . . : YES      (Yes/No)

Utility control options:
 Generate template statements . . . : NO       (Yes/No)

BP - Change batch job parameters
TU - Specify template usage
```

Figure 5-41 Migrate Parameters panel (2/2)

When you press ENTER, the jobs that are generated are identical to a migration of a single physical database as discussed in 5.4.2, “Creating (cloning) development databases (D1-D4) using MIG” on page 213.We get two jobs, MADDS1 and MADDSE, as shown in Figure 5-42.

```
 Menu  Functions  Utilities  Help
ssssss
```

```
DB2 Admin ---------------- Edit Statement ---------------- Columns 00001 00072
Command ===>                                                  Scroll ===> CSR

****** ***************************** Top of Data ******************************
000001 -- Created by PAOLOR2 on 2007/03/05 at 15:45 by cloning of
000002 -- source work stmt list  QMADD  from library  'ADMIN.CLONE.WSL'
000003 -- source work stmt list  MADD  from library  'ADMIN.MIG.WSL'
000004 -- START OF ADD STATEMENTS BY ADB2A2WL EXEC
000005 -- START OF MIGRATE WORK STATEMENTS
000006 --
000007 -- THE FOLLOWING DATA SETS MUST BE AVAILABLE ON THE SYSTEM WHERE THIS
000008 -- WORK STATEMNT LIST IS TO BE RUN. THEY SHOULD NOT BE DELETED AS LONG
000009 -- AS THIS WORK STATEMENT LIST IS TO BE USED.  WHEN TRANSFER THIS WORK
000010 -- STATEMENT LIST TO ANOTHER SYSTEM, MAKE SURE THOSE DATA SETS ARE
000011 -- ALSO TRANSFERRED.
000012 --
000013 --
000014 -- DB2 UNLOAD OUTPUT/LOAD INPUT DATA:
000015 --
000016 --   PAOLOR2.MADD.ULD.S1
000017 --   PAOLOR2.MADD.ULD.S2
000018 --   PAOLOR2.MADD.ULD.S3
000019 --   PAOLOR2.MADD.ULD.S4
000020 --   PAOLOR2.MADD.ULD.S5
000021 --   PAOLOR2.MADD.ULD.S6
000022 --   PAOLOR2.MADD.ULD.S7
000023 --   PAOLOR2.MADD.ULD.S8
000024 --   PAOLOR2.MADD.ULD.S9
000025 --   PAOLOR2.MADD.ULD.S10
000026 --   PAOLOR2.MADD.ULD.S11
000027 --   PAOLOR2.MADD.ULD.S12
000028 --   PAOLOR2.MADD.ULD.S13
000029 --   PAOLOR2.MADD.ULD.S14
```

Figure 5-44 Work Statement List Edit mode - Unload data set names

As you scroll down the edited WSL data set, you can find the relationship between the unloaded data set names and table spaces as shown in Figure 5-45.

```
DB2 Admin ---------------- Edit Statement ---------------- Columns 00001 00072
Command ===>                                                  Scroll ===> CSR

000042 -- RELATION BETWEEN THE LAST PART OF THE GENERATED DATA SETS NAMES
000043 -- AND DB2 TABLE SPACES:
000044 --
000045 --    DATASET NAME   TABLESPACE NAME
000046 --    ------------   -----------------
000047 --
000048 --    ...S1          DSG24P01.GLWSDPT
000049 --    ...S2          DSG24P01.GLWSEMP
000050 --    ...S3          DSG24P01.GLWSEPA
000051 --    ...S4          DSG24P01.GLWSPJA
000052 --    ...S5          DSG24P01.GLWSPRJ
000053 --    ...S6          DSG24P01.GLWSSPL
000054 --    ...S7          DSG24P01.GLWS001
000055 --    ...S8          DSG24P01.GLWS002
```

Figure 5-45 Work Statement List Edit mode - Data set names and table space names

And then finally the LOAD statement refers to these unload data set names as shown in Figure 5-46.

```
004347 -- BEGIN OF ADB28RL1
004348 TEMPLATE SYSREC DSN PAOLOR2.MADD.ULD.S1
004349                 DISP(SHR, KEEP, KEEP);
004350 TEMPLATE SYSERR DSN PAOLOR2.MADD.TEST.SYSERR
004351                 DISP(MOD, DELETE, CATLG)
004352                 UNIT SYSDA
004353                 SPACE(2, 1) TRK;
004354 TEMPLATE SYSUT1 DSN PAOLOR2.MADD.TEST.SYSUT1
004355                 DISP(MOD, DELETE, CATLG)
004356                 UNIT SYSDA
004357                 SPACE(2, 1) TRK;
004358 TEMPLATE SORTOUT DSN PAOLOR2.MADD.TEST.SORTOUT
004359                 DISP(MOD, DELETE, CATLG)
004360                 UNIT SYSDA
004361                 SPACE(2, 1) TRK;
004362 TEMPLATE SYSMAP DSN PAOLOR2.MADD.TEST.SYSMAP
004363                   DISP(MOD, DELETE, CATLG)
004364                   UNIT SYSDA
004365                   SPACE(2, 1) TRK;
004366 LOAD DATA INDDN SYSREC LOG NO RESUME YES
004367  ENFORCE NO
004368  EBCDIC CCSID(00037, 00000, 00000)
004369  INTO TABLE "DSG24P21"."GLWTDPT"
004370  WHEN(00001:00002) = X'0003'
004371  ("DEPT_NO"
004372         POSITION(00003:00006) INTEGER
004373  , "DEPT_NAME"
```

Figure 5-46 WSL Edit mode - LOAD statement with unloaded data set names and table spaces

If the object names in the source and target environment are different, using the DB2 Compare technique might be a better solution to migrate your objects.

Using DB2 Compare

In this section we use a similar technique as we do for updating a database design, described in 6.2, “Modifying the Master Development Database (D0) using generated DDL and Object Compare” on page 292. The only difference is that there are no target databases for comparison in this case because this is the initial roll out of a design (as shown in Figure 5-47).

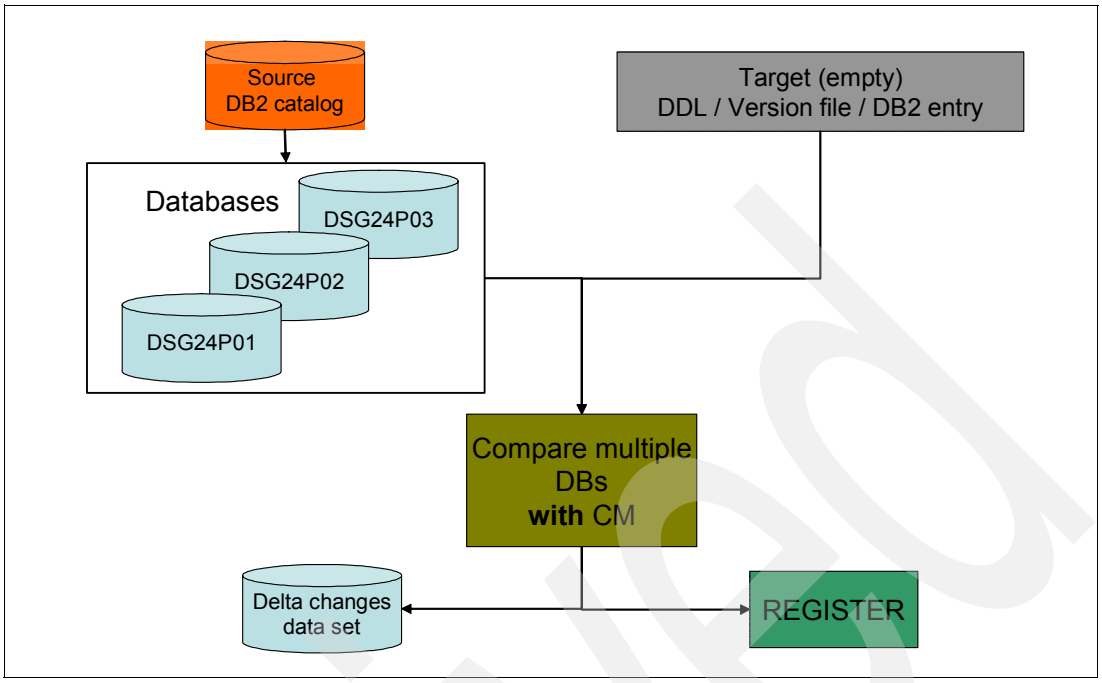

Figure 5-47 Cloning with Compare against empty target

If this is your first encounter with DB2 Compare, you can read “REGISTER using Compare” on page 139. It contains a lot of general information about how DB2 Compare works.

Using the panels

We now go through the five numbered steps in the DB2 Compare Menu (see Figure 5-48).

```
GOCMENU ---------------- DB2 Object Comparison Tool Menu ---------------- 17:42
Option ===>

                                        Status:
   1 - Specify compare source (new)     DB2 catalog extract specified
   2 - Specify compare target (old)     DB2 catalog extract specified
   3 - Specify compare masks            Mask specified
   4 - Specify fields to ignore         IGNORE Data Set Specified
   5 - Generate compare job             Generated

   W - Walk through steps 1 - 5 in sequence
   V - Generate job to extract version file from source only
   R - Reset status

   S  - Save dialog
   M  - Manage/Restore dialog
   MC - MultiCompare
```

Figure 5-48 DB2 Compare Menu

1 - Specify compare source (new)

After choosing *Specify compare source (new)* on the DB2 Compare Menu, and selecting *Source is from the DB2 catalog* from the Specify compare source panel, we fill in our data set name and comment on the Specify DB2 Source Catalog Extract panel, as shown in Figure 5-49.

```
GOC12 e -------------- Specify DB2 Source Catalog Extract --------------- 19:24
Option ===> 1


   1 - Source is databases from the DB2 catalog
   2 - Source is table spaces from the DB2 catalog
   3 - Source is tables from the DB2 catalog
   4 - Add schema objects to the DB2 Source catalog extract

Specify compare version file output:
 Version table entry:
   Owner . . .  PAOLOR2  >              (? to look up)
   Name  . . .  DSG24POX V0        >    (? to look up)
 Data set:
  Data set name . .

Enter a description (optional):

  Description  . . THIS IS VERSION 0
```

Figure 5-49 Specify DB2 Source Catalog Extract

We then choose option **1** which brings us to the Specify Source DB2 Databases panel where we enter an **I** besides the empty entry (see Figure 5-50).

```
GOC1D e ----------------- Specify Source DB2 Databases ------------ Row 1 of 1
Command ===>                                              Scroll ===> PAGE

Commands: RESET
Line commands:
 D - Delete  I - Insert

Select Database  Location
       *         *
------ --------  ----------------
i      ........  ................
****************************** END OF DB2 DATA ******************************
```

Figure 5-50 Specify Source DB2 Databases panel

This brings us to the Compare: Add Databases panel (see Figure 5-51).

```
GOC1DA  ------------------ DB8A Compare Add Databases ------------------ 19:28
Option ===>

Enter the partial name of the database you want to add to the compare
operation:

  Partial database name  . . : DSG24PO
  Location name  . . . . . . : DB8A            >

Press enter to search for the database.
```

Figure 5-51 Compare Add Databases panel 1/2

There we specify the source database name **DSG24PO** and press Enter, which brings us to the following panel (see Figure 5-52).

```
 GOC1DD  -------------------- DB8A Compare Add Databases ----------- Row 1 of 4
Command ===>                                                  Scroll ===> PAGE

Valid line commands are:                          Location: DB8A
 S - Select (add)

Select Database Action
       *        *
------ -------- ------
       DSG24P0
s      DSG24P01
s      DSG24P02
s      DSG24P03
```

Figure 5-52 Compare Add Databases panel 2/2

We place an **S** in front of all the physical databases we are interested in, and press Enter. This results in the selection of those databases (Action Added) as shown in Figure 5-53.

```
GOC1DD  -------------------- DB8A Compare Add Databases ----------- Row 1 of 4
Command ===>                                                  Scroll ===> PAGE

Valid line commands are:                          Location: DB8A
 S - Select (add)

Select Database Action
       *        *
------ -------- ------
       DSG24P0
*      DSG24P01 Added
*      DSG24P02 Added
*      DSG24P03 Added
****************************** END OF DB2 DATA ******************************
```

Figure 5-53 Compare Add Databases - after selection

Now we press PF3 which brings us back to the Specify Source DB2 Databases panel with our selected databases listed (see Figure 5-54).

```
 GOC1D e ---------------- Specify Source DB2 Databases ------------ Row 1 of 4
Command ===>                                                  Scroll ===> PAGE

Commands: RESET
Line commands:
 D - Delete  I - Insert

Select Database  Location
       *         *
------ --------  ----------------
*      ........  ................
       DSG24P01  DB8A
       DSG24P02  DB8A
       DSG24P03  DB8A
****************************** END OF DB2 DATA ******************************
```

Figure 5-54 Specify Source DB2 Databases - After selection

Press PF3 to return to the DB2 Compare Menu.

Archived

2 - Specify compare target (old)

We can now start selecting *Specify compare target (old)* on the DB2 Compare Menu. This takes us to the Specify Compare Target panel, where we select **2**, which takes us to Specify DB2 Target Catalog Extract panel (see Figure 5-55).

```
 GOC12 e -------------- Specify DB2 Target Catalog Extract --------------- 19:33
Option ===>


   1 - Target is databases from the DB2 catalog
   2 - Target is table spaces from the DB2 catalog
   3 - Target is tables from the DB2 catalog
   4 - Add schema objects to the DB2 Target catalog extract

Specify compare version file output:
 Version table entry:
   Owner . . .  PAOLOR2  >              (? to look up)
   Name  . . .  NULL VERSION       >    (? to look up)
 Data set:
  Data set name . .

Enter a description (optional):

  Description  . . NO OBJECT - FOR INITIAL CLONE
```

Figure 5-55 Specify DB2 Target Catalog Extract

After entering the appropriate Version data set name and Description, we enter **1** to select *Target is databases from the DB2 catalog* and press **Enter**, which takes us to the Specify Target DB2 Databases panel (see Figure 5-56).

```
 Compare ---------------- Specify Target DB2 Databases ------- Row 1 to 1 of 1
 Command ===>                                              Scroll ===> PAGE

 Valid commands are: RESET
 Valid line commands are:
  D - Delete  I - Insert

 Select Database  Location
        *         *
 ------ --------  ----------------
        ........  ................
 ****************************** END OF DB2 DATA ******************************
```

Figure 5-56 Specify DB2 Target Databases

Since we are cloning the source databases for the first time, the database does not yet exist, so we leave this panel empty and press PF3. This returns us to the DB2 Compare Menu.

Note: If this panel is not empty, type **RESET** on the command line and press **Enter.**

3 - Specify compare masks

We are now ready to *specify compare* masks by selecting option **3** (see Figure 5-57).

```
GOC3 re --------------------- Specify Compare Masks ---------------------------
Option ===>


Mask Table Entry:
  Owner . .  PAOLOR2  >            ( to look up)
  Name  . .  ?                  >  ( to look up)
Data Set:
  Mask DSN  . .
Options:
  Edit Mask . .  NO  (Yes/No)
```

Figure 5-57 Specify Compare Masks

We have created a mask for this in a previous example but cannot remember the name. So we look it up with a **?**. Pressing Enter takes us to an overview of all masks with owner PAOLOR2 as shown on the next panel (see Figure 5-58).

```
ADB2C21 n ------------------- CM - Masks -------------------------- Row 1 of 5
Command ===>                                                Scroll ===> PAGE
Select by typing '+'
Line commands:
 U - Update  DEL - Delete  INS - Insert  ML - Mask lines  CH - Changes
 E - Edit  I - Details on mask

Sel          ID Owner    Name                Comment
              * *        *                   *
--- ----------- -------- ------------------ --------------------------------->
             21 PAOLOR2  DSG24DOG-DSG25DOP  MASKING FROM DATABASE DSG24DOG TO D
+            50 PAOLOR2  DSG24PO_TO_DSG24P1 MASKING FOR PO TO P1 DATABASES
             51 PAOLOR2  DSG24P1_TO_DSG24P2
              1 PAOLOR2  PAOLOR2-PAOLOR2Z
             31 PAOLOR2  SEW2
****************************** END OF DB2 DATA ******************************
```

Figure 5-58 Specify CM Masks and select with “+”

Pressing Enter takes us to the next panel, where we see the content of the selected mask and we can verify whether the mask definition is appropriate. See Figure 5-59.

```
ADB2C2L n ----------------- CM - Mask Lines ----------------------- Row 1 of 2
Command ===>                                                Scroll ===> PAGE

Mask lines for mask "PAOLOR2"."DSG24PO_TO_DSG24P1"
Commands: SAVE
Line commands:
  I - Insert  D - Delete  R - Repeat  M - Move  A - After  B - Before

Sel    Sequence Type     From                  To                     Oper.  T
              * *        *                     *                      *      *
--- ----------- -------- --------------------> ---------------------> ------ -

              1 DBNAME   DSG24PO*              DSG24P1*
              2 AUTHID   DSG24PO*              DSG24P1*
****************************** END OF DB2 DATA ******************************
```

Figure 5-59 CM Mask Lines panel

This tells Compare to convert the names of all of our database from **DSG24P0**, to **DSG24P1**. The ***** at the end is a wildcard which makes sure the last character of the target is the same as the source.

> **Note:** When determining your naming convention, be sure to leave enough positions at the end to accommodate for having one database for each of your table spaces. We only use one position which might not be enough for some applications.

4 - Specify fields to ignore

Pressing PF3 returns us to DB2 Compare Menu. Since we do not have to ignore anything in the Compare (since the target is empty) we can skip the option **4**. If you want to work with ignore, then you can provide a generic entry (see Figure 5-60).

```
GOC4 re ----------------- Specify Compare Ignore Fields -----------------------
Option ===>



Ignore Table Entry:

  Owner . . PAOLOR2  >                (? to look up)
  Name  . . ?                  >     (? to look up)
Data Set:
  Data Set Name . .
Options:
  Edit Ignores  . . YES (Yes/No)
```

Figure 5-60 Specify Compare Ignore Fields lookup with "?"

Here also we are not sure which ignore table entry we should use, so we type **?** in the name field. See Figure 5-61.

```
ADB2C31 n ------------------ CM - Ignores ------------------------- Row 1 of 2
Command ===>                                                  Scroll ===> PAGE
Select by typing '+'
Line commands:
 U - Update  DEL - Delete  INS - Insert  IL - Ignore lines  CH - Changes
 I - Details on ignore

Sel          ID Owner    Name                Comment
              * *        *                   *
--- ---------- -------- ------------------ ----------------------------------
              1 PAOLOR2  EXCLUDE CREATOR
+            21 PAOLOR2  NULL VERSION       FOR COMPARING TO NULL FILE
****************************** END OF DB2 DATA ******************************
```

Figure 5-61 CM Ignores can be selected with "+"

Our selected ignore entries are shown on Figure 5-62.

```
GOCCI e ---------------- Specify Ignore Fields: Objects ---------- Row 1 of 18
Command ===>                                                   Scroll ===> PAGE

Valid line commands are:
 U - Update Ignore Fields

Select Object          Ignore Fields
       *               *
------ --------------- ---------------------------------------------------------
       GENERIC         None
       SYSCHECKS       None
       SYSCOLUMNS      None
       SYSDATABASE     None
       SYSDATATYPES    None
       SYSFIELDS       None
       SYSINDEXES      None
       SYSINDEXPART    None
       SYSKEYS         None
       SYSPARMS        None
       SYSRELS         None
       SYSROUTINES     None
       SYSSEQUENCES    None
       SYSTABLEPART    None
       SYSTABLES       None
```

Figure 5-62 Specify Ignore Fields panel

Since there are no objects currently in the target databases, we ensure that all Ignore Fields are `None`.

Tip: In this case there are no rows in the ignore table ADBIGNORES. However, there is an entry in table ADBIGNORE.

5 - Generate Compare jobs

Pressing **PF3**, returns us to the DB2 Compare Menu, where we type **5** and press Enter to take us to the Generate Compare Jobs panel (see Figure 5-63).

Archived

```
GOC5 re -------------------- Generate Compare Jobs  ---------------------------
Option ===>

 Specify the following for DB2 Object Comparison Tool:
                                                                    More:     +
 Worklist information:
   Worklist name . . . . . . . COMP     (also used as middle qualifier in DSNs)

 Compare options:
   Suppress DROP of objects  . NO       (Yes/No)
   Suppress DROP of columns  . NO       (Yes/No)
   Suppress adding columns . . NO       (Yes/No)
   Run SQLID . . . . . . . . . <NONE>   (Blank, an SQLID, or <NONE>)

 Change reporting options  . . no       (Yes/No)

 Data set information:
   PDS for batch jobs  . . . . 'ADMIN.JCL'
   Prefix for data sets  . . . PAOLOR2
```

Figure 5-63 Generate Compare Jobs panel 1/3

After filling in all the fields shown in Figure 5-63, we press PF8 to scroll down (see Figure 5-64).

```
GOC5 re -------------------- Generate Compare Jobs  ---------------------------
Option ===>

 Specify the following for DB2 Object Comparison Tool:
                                                                    More:   - +
 Options:
   Single compare job  . . . . YES      (Yes/No)
     Member name . . . . . . . COMP     (default COMPARE )
   Generate apply jobs . . . . YES      (Yes/No)
     Generate one job  . . . . YES      (Yes, No, or (Per) Process)
       Member prefix . . . . .          (default APPLY )
     As work statement list  . YES      (Yes/No to append to work stmt list)
     Use utility options . . .          (Yes/No)
     Unload Method . . . . . . U        (Unload, Parallel unload)
     Generate templates. . . . NO       (Yes/No)
```

Figure 5-64 Generate Compare Jobs panel 2/3

We specified to create a work statement list and want one job generated. When we specify Generate apply jobs = **YES**, then we want to get the changes generated, which means in turn that this action kicks off Change Management later.

But first, let us use PF8 to scroll down the panel and complete the remaining options as shown in Figure 5-65.

```
 Optional jobs after Reload or Alter:
        Run CHECK DATA  . . . . NO       (Yes/No)
        Take an image copy  . . N        (after: Reload/Alter/Both/None)
        Run REORG/REBUILD . . . N        (Mandatory, All relevant, None)
        Run RUNSTATS  . . . . . N        (after: Reload/Alter/Both/None)
        Run REBIND  . . . . . . NO       (Yes/No)

  BP - Change batch job parameters
  TU - Specify TEMPLATE usage
  UO - Change utility options
```

Figure 5-65 Generate Compare Jobs panel 3/3

Since there is no data involved in this operation, we entered **N** for the Run CHECK DATA, Run RUNSTATS, Take an image copy, Run REORG, and Run REBIND fields. Pressing Enter pops up change management — remember we asked for Generate apply jobs = **YES**. So we then specify **Yes** to use change management for this function (see Figure 5-66).

```
EssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
e ADB2CMRO  ----------- DB8A Change Management Prompt ----------- 20:28 e
e                                                                      e
e Change Management is optional for SQLID:  PAOLOR2                    e
e                                                                      e
e Do you wish to use Change Management for this function: yes (Yes/No)  e
e                                                                      e
e                                                                      e
e                                                                      e
DssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

Figure 5-66 Change Management Prompt panel

Remember — we only get this pop-up panel when the CM default setting is optional (either to the specific userid PAOLOR2 or system wide). After pressing Enter we get to the Change Management register options (see Figure 5-67).

```
DB2 Admin -------------------- CM - Register Options -------------------- 20:42
Option ===>

Commands: CONTINUE                                      DB2 System: DB8A
                                                        DB2 SQL ID: PAOLOR2
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR2  >  (Optional, default is PAOLOR2)
Name  . . . . . . . . COMP                                            >
Comment . . . . . . . COMPARE AGAINST NULL FILE                       >
Change Type . . . . : CHANGE   (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                     Owner         Name
Ignore  . . . . . . .         >                           >  (? to look up)
Mask  . . . . . . . .         >                           >  (? to look up)
Delta Version . . . .         >                           >  (? to look up)
```

Figure 5-67 Change Management Prompt panel

Archived

Now we can provide a meaningful name and a helpful comment so that later it gets easier to identify where this change came from and what is it about. All the other fields can be left blank for Ignore, Mask, and Delta Version. Then we position our cursor on **CONTINUE** and press Enter; this brings us to the next pop-up panel (see Figure 5-68).

Attention: If we just press **Enter,** without specifying **CONTINUE** then the panel stays and fills in a default value for Delta Version, in our case it would be:

Delta Version . . . . PAOLOR2 > COMP > (? to look up)

So we only get further with **CONTINUE.**

```
EssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
e Compare ----------------- Specify Register Mode ----------------- 20:57 e
e                                                                           e
e Action if pending changes  PREREQ    (Cancel/Prereq/Supersede)            e
e                                                                           e
e                                                                           e
DsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

Figure 5-68 Specify Register Mode

Now we get asked what shall happen to pending changes. There should not be any pending changes, since this is supposed to be the initial clone. However, even if pending changes do exist, we might not care about this because we register the change on the current DB2 (DB8A) but the actual rollout can be on another DB2. The intention is to use the outcome of the COMPARE process which is the delta files. These delta files are then imported on the target DB2.

So when confirming the PREREQ option for pending changes by pressing Enter, we get a generated job which does the Compare. The result is a delta file and the registration of the change in the CM database. If we look at the sequence of steps we can see what goes on behind the scenes (see Figure 5-69).

```
 File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
ssssss
```

Then from here we can go in two different directions, as shown in Figure 5-71.

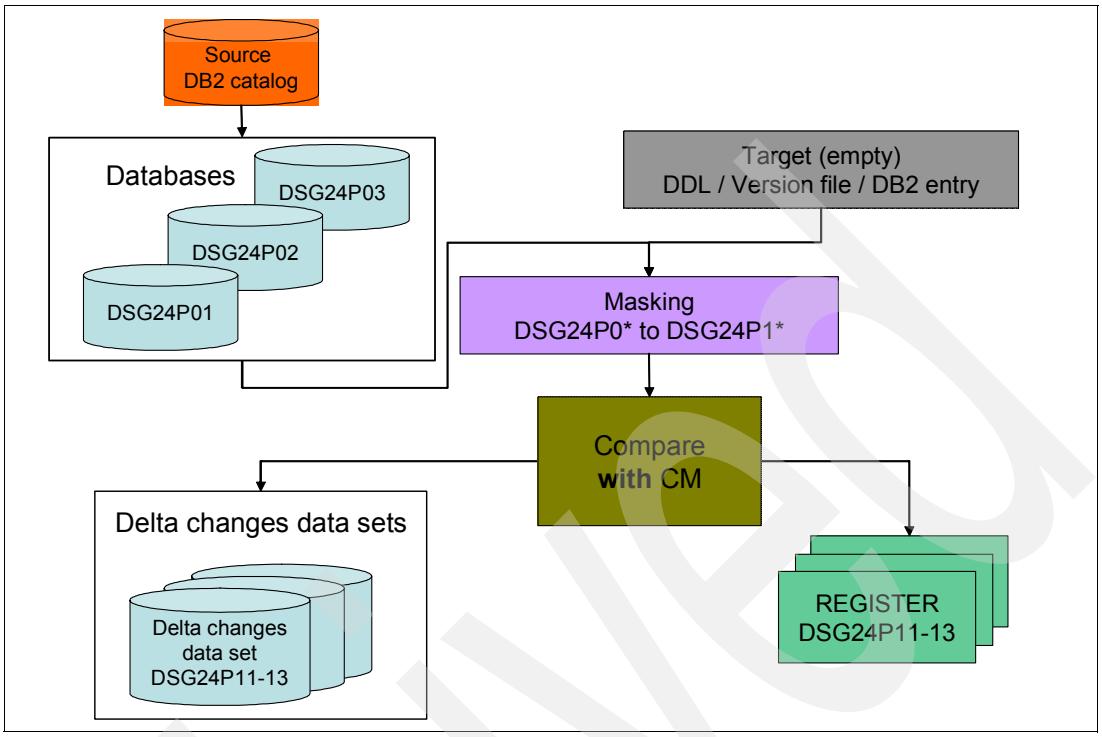

Figure 5-71 Cloning multiple databases with new CM function

We can either:

- Stay on the local DB2:
  - It already got REGISTERED from the COMPARE step
  - ANALYZE
  - RUN
- Go to another DB2:
  - Take the DELTA file to the target environment
  - IMPORT the DELTA file (which gets the CHANGE registered)
  - You can use appropriate masking during the IMPORT, although we already did the correct masking during the COMPARE.
  - ANALYZE
  - RUN

This was the Compare method exploiting the new CM functions.

Note: The Compare technique allows the migration of multiple databases in a single procedure, but does not migrate any data.

Executing and cloning using work statement lists (WSL)

In the scenario we described above, we did use the new CM functions for the Compare. Since initial cloning does not necessarily require change control to be in place, we could have also worked without CM functions when doing the Compare to get the proper DDL generated.

By specifying `N` for using the CM function we could have created a work statement list outside of the normal CM process. Once a WSL is created, you execute it, or clone it before executing it. Cloning is an efficient technique to change object names for different environments.

Executing and cloning a WSL is described in 3.9.6, “Executing a work statement list” on page 104 and in 3.9.7, “Cloning a work statement list” on page 105.

IMPORT to create the new database on the target environment

The import scenario describes how one can take the delta file which has been produced by the previous Compare scenario. We bring the delta file to the target environment where the database does not exist. Then we use the IMPORT process to register the definitions of the clone in the CM database of the target environment (see Figure 5-72).

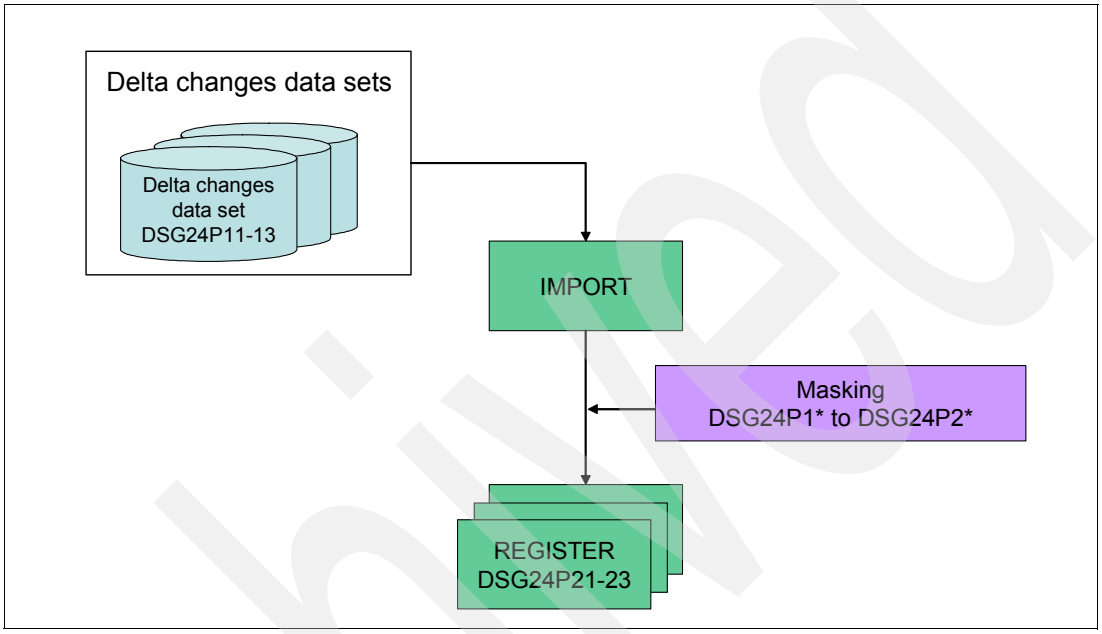

Figure 5-72 Overview on import to process delta or DDL files

We require the masking if we want to change the names of the objects in the delta or DDL file.

> **Tip:** We could have also used a plain DDL file that comes out of the GEN or any other DDL generation process.

So we start with the delta file to be imported and first we get to the Manage Changes panel (see Figure 5-73.

```
DB2 Admin ---------------------  CM - Manage Changes -------------------- 14:51
Option ===> 4


   1 - Display changes                                   DB2 System: DB8A
   2 - Create a change                                   DB2 SQL ID: PAOLOR2
   3 - Create delta for target
   4 - Import changes
```

Figure 5-73 CM Changes panel

We select **4** and proceed to the next step, which allows us to define the data set which is the input for the IMPORT process (see Figure 5-74).

```
DB2 Admin ------------------- CM - Import Changes ----------------------- 14:55
Command ===>


Input data set information:                                DB2 System: DB8A
 Data set name 'PAOLOR2.COMP.CHANGES'

Prereq resolution mode  . B         (TSO/Batch)
Execution mode  . . . . . B         (TSO/Batch)
```

Figure 5-74 Import Changes panel

This input data set can either be DDL or a DELTA file. In our case we use the delta file which was produced by the last Compare scenario. Our worklist name was **COMP.** So we find the delta file under the following name : *userid.worklist*.CHANGES.

Importing a change is a two-phase process in which DB2 Admin first determines if any prerequisite changes are pending for the affected objects and then registers the change. This processing can be done in the foreground, the background, or a combination of both. We can now specify how we want to deal with pending prerequisites (see Figure 5-75).

```
EsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
e DB2 Admin -- DB8A CM Import a Change - Action for Pending Changes - 15:07 e
e                                                                          e
e Specify an action to use if the imported change has pending prerequisite  e
e                                                                          e
e                                                                          e
e                                                                          e
e                                                                          e
e Select a choice                                                          e
e 1 1.  Cancel                                                             e
e   2.  Prereq                                                             e
e   3.  Supersede                                                          e
e                                                                          e
e                                                                          e
e                                                                          e
e                                                                          e
DsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

Figure 5-75 Import a Change - Action for Pending Changes panel

Archived

We choose **1** (Cancel) because no dependencies (pending changes) should exist for a new database. Then we go ahead and get to the registration panel (see Figure 5-76).

```
DB2 Admin -------------------- CM - Register Options -------------------- 15:12
Option ===>

Commands: CONTINUE                                        DB2 System: DB8A
                                                          DB2 SQL ID: PAOLOR2
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR2  >  (Optional, default is PAOLOR2)
Name  . . . . . . . . IM01                                                >
Comment . . . . . . . IMPORT NEW DATABASE FROM DELTA FILE                 >
Change Type . . . . : PROMOTE  (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                      Owner         Name
Ignore  . . . . . . .         >                           >  (? to look up)
Mask  . . . . . . . .         >                           >  (? to look up)
Delta Version . . . . PAOLOR2  >    IM01                 >  (? to look up)
```

Figure 5-76 Register Options panel

We specify a meaningful name (IM01) and comment. Then we enter CONTINUE.

Tip: If you look at the “Change Type” you see PROMOTE. This change type becomes CHANGE as soon as the register is done. This can be confusing in the beginning; however, there are attempts underway to correct this naming.

This generates a job and we submit it to get the delta file imported. The job ends with RC=28 and we wonder why this happened.

```
15.32.08 JOB21987  IEF403I PAOLOR2I - STARTED - TIME=15.32.
15.32.20 JOB21987  -
15.32.20 JOB21987  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP
15.32.20 JOB21987  -PAOLOR2I          S02IMP      28   2775
```

Figure 5-77 Joblog of failed IMPORT

Further down in the job output we discover the reason for the error. Obviously there are dependencies that we did not expect (see Figure 5-78).

```
ADB2CID - Pending Changes

Change ID Owner    Name               STMT
--------- -------- ------------------ --------------------------------
       59 PAOLOR1  D070306-CR#20-NOR  CREATE INDEX DSG24D01.GLWXEPA4
```

Figure 5-78 Pending Changes

Then we go into CM again and check this change ID 59 and want to see its related objects (see Figure 5-79).

```
DB2 Admin ------------------ CM - Changes ------------------------- Row 1 of 1
Command ===>                                                   Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- ---------------------
SR        59 PAOLOR1  D070306-CR#20-NOR  CHANGE  DEFINED
****************************** END OF DB2 DATA ******************************
```

Figure 5-79 Changes panel

The **SR** command gives an overview of the related objects. We hope this gives us a clue why our IMPORT got cancelled (see Figure 5-80).

```
DB2 Admin -- Browse Result of SQL Select              ------------ Row 1 of 3
Command ===>                                                   Scroll ===> PAGE

L CHANGEID   CHANGESEQ OBJECTTYPE QUALIFIER NAME     BASEIND
         *          * *          *         *        *
- -------- ---------- ---------- --------- -------- -------
        59           1 IX         DSG24D01  GLWXEPA4       1
        59           1 SG                   GLWG01         0
        59           1 TB         DSG24D01  GLWTEPA        0
****************************** END OF DB2 DATA ******************************
```

Figure 5-80 Show related objects for the change

From looking at the objects we see that there is an index and table which do not relate to our new database. However, this index uses storage group GLWG01 and this is also used by our new database. This is why a pending change is detected. But we want to supersede this pending prerequisite (see Figure 5-81).

```
EssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
e DB2 Admin -- DB8A CM Import a Change - Action for Pending Changes - 15:07 e
e                                                                          e
e Specify an action to use if the imported change has pending prerequisite  e
e                                                                          e
e                                                                          e
e                                                                          e
e                                                                          e
e Select a choice                                                          e
e 3 1.  Cancel                                                             e
e   2.  Prereq                                                             e
e   3.  Supersede                                                          e
e                                                                          e
e                                                                          e
e                                                                          e
e                                                                          e
DsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

Figure 5-81 Import a Change - Action for Pending Changes panel

After this choice we got another IMPORT job generated, submitted it again, and this time it ended with RC=0. We see this also confirmed on our changes panel (Figure 5-82).

```
DB2 Admin ------------------ CM - Changes ------------------------ Row 1 of 101
Command ===>                                                 Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name               Type    Status   Comment
           * *        *                  *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
AN       237 PAOLOR2  IM01               CHANGE  DEFINED  IMPORT NEW DATABASE F
```

Figure 5-82 Changes panel

The change type has changed to CHANGE (from IMPORT) and the status is DEFINED. We can now analyze the change. When the ANALYZE completes successfully, we see the status of the change is ANALYZED (as shown on the Change panel in Figure 5-83).

```
DB2 Admin ------------------ CM - Changes ----------------------- Row 1 of 101
Command ===>                                                  Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
RN       237 PAOLOR2  IM01                CHANGE  ANALYZED IMPORT NEW DATABASE F
```

Figure 5-83 Changes panel

We now can RUN the change and generate the job by typing RN in front of the change. This brings us to the RUN panel (see Figure 5-84).

```
DB2 Admin ---------------- CM - Run a Change ---------------------------------
Command ===>


Change . . . : PAOLOR2.IM01

                                                                  More:     +
Data set information:
  PDS for batch jobs . . . . . 'ADMIN.CM.JCL'
  Prefix for data sets . . . . PAOLOR2

Create a new base version  . . NO   (Yes/No)

  Scope Information:
    Owner  . . . . . . . . . .         >           (? to look up)
    Name . . . . . . . . . . .                   >  (? to look up)

  Version Information:
    Owner  . . . . . . . . . .         >           (? to look up)
    Name . . . . . . . . . . .                   >  (? to look up)

Runtime analyze:
  Change reporting options . . NO   (Yes/No)
```

Figure 5-84 Run a change panel

We specified our JCL library (**'ADMIN.CM.JCL'**) and said that we do not want a new base version created (since we have not yet set up a version scope for that database), although it might be good practice to establish a base version (v0) after creating a new database.

We press Enter. This generates our job, and we submit the job. When it finishes, we see the status of the change as COMPLETED. Then we check whether all the databases have been built (see Figure 5-85).

```
DB2 Admin ------------------- DB8A Databases ---------------------- Row 1 of 3
Command ===>                                                  Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                         Storage  Buffer          Created      Index
Select Name     Owner    Group    Pool       DBID By       T E Buffer Pool
       *        *        *        *             * *        * * *
------ -------- -------- -------- -------- ------ -------- - - -----------
       DSG24P11 PAOLOR2  GLWG01   BP15        311 PAOLOR2    E BP16
       DSG24P12 PAOLOR2  GLWG01   BP15        312 PAOLOR2    E BP16
       DSG24P13 PAOLOR2  GLWG01   BP15        313 PAOLOR2    E BP16
```

Figure 5-85 Databases panel showing the new databases

We can now repeat this process to rollout this new databases in every target environment. If these target environments require different naming conventions then we can use masks on the IMPORT process (see Figure 5-86).

```
DB2 Admin -------------------- CM - Register Options -------------------- 18:28
Option ===>

Commands: CONTINUE                                  DB2 System: DB8A
                                                    DB2 SQL ID: PAOLOR2
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR2  >  (Optional, default is PAOLOR2)
Name  . . . . . . . . IMO2                                            >
Comment . . . . . . . IMPORT NEW DATABASE FROM DELTA FILE WITH MASK   >
Change Type . . . . : PROMOTE  (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                      Owner        Name
Ignore  . . . . . . .         >                          >  (? to look up)
Mask  . . . . . . . . PAOLOR2  >    DSG24P1_TO_DSG24P2   >  (? to look up)
Delta Version . . . .         >                          >  (? to look up)
```

Figure 5-86 Register the import with mask

The mask itself translates the **DSG24P1*** databases (**DSG24P11**, **DSG24P12**, **DSG24P13**) into the **DSG24P2*** databases (**DSG24P21**,**DSG24P22**, **DSG24P23**). The mask can be seen below in Figure 5-87.

```
DB2 Admin ----------------- CM - Mask Lines ----------------------- Row 1 of 2
Command ===>                                                   Scroll ===> CSR

Mask lines for mask "PAOLOR2"."DSG24P1_TO_DSG24P2"
Commands: SAVE
Line commands:
  I - Insert  D - Delete  R - Repeat  M - Move  A - After  B - Before

Sel    Sequence Type     From                  To                     Oper.  T
              * *        *                     *                      *      *
--- ----------- -------- --------------------> ---------------------> ------ -

              1 DBNAME   DSG24P1*               DSG24P2*
              2 AUTHID   DSG24P1*               DSG24P2*
```

Figure 5-87 Mask lines panel

The IMPORT process must be completed for every target environment because appropriate masking must be specified for each target environment with different naming conventions.

So we have shown various methods to rollout the initial version of a new database. This means the source environment and all the target environments should have exactly the same structures. Any comparison at this stage should not show any differences.

So now we can move on and start to modify the source environment. This is described in Chapter 6, “Implementing changes to a design (Vn)” on page 257.

Archived

6

Implementing changes to a design (Vn)

In this chapter we give you the details of how you would implement changes to the data base design. An important first step is creating a base version file. One way of modifying the database objects is to use Alter Column (ALC). Another is Alter (AL). We go through these processes in detail using CM. Another way of making changes to a database is to use Object Compare, and we go through similar types of changes using this method.

We cover the following topics:

- Modifying DB2 objects using DB2 Admin (ALC) and (AL)
- Modifying the Master Development Database (D0) using from generated DDL and Object Compare

© Copyright IBM Corp. 2007. All rights reserved.

6.1 Modifying DB2 objects using DB2 Admin (ALC) and (AL)

In this section we review the steps to modify the DB2 objects using DB2 Admin ALC and AL commands.

6.1.1 Creating a base version file (required)

A version is a snapshot of the definitions of a set of objects at a specific time. These objects normally represent the objects of an application or application area. There are two types of version files:

- Base version
- Delta version

It is very important to create a base version file at the time the objects are first defined or created. A base version allows you to track changes to the set of objects, restore objects to a previous point in time if fall back is required and promote the changes from one system (such as a development system) to another system (such as test or production).

Important: We highly recommend that all versions be stored in the Change Management database. This allows easier tracking, access, and recovery.

See Chapter 4, “Working with the Change Management functions” on page 111 for all the steps for creating a base version.

Once there is a base version, you can review who created the version, when it was created, what the version scope consists of, and if necessary, you can update, insert, promote, or delete the base version.

An example of the base version for database **DSNG24D04** is in Figure 6-1.

```
DB2 Admin ------------------ CM - Versions ----------------- Row 1 to 31 of 97
Command ===>                                                  Scroll ===> PAGE

Line commands:
 CH - Changes  PR - Promote  VS - Version scope  DEL - Delete  U - Update
 PT - Toggle protected status  I - Details on version

Sel          ID T Owner    Name                Comment
              * * *        *                   *
--- ---------- - -------- -----------------> -------------------------------
            178 B PAOLOR4  BASE1DSG24D04      BASE VER1 DB DSNG24D04 ALL - DB
```

Figure 6-1 CM Version panel (ADB2C41)

Important: For important base versions, we recommend that you issue the **PT** command on CM Version panel, to protect them. This should be done on an individual basis.

To see the details for this base version, issue the **I** command from CM Version panel. See Figure 6-2.

```
DB2 Admin ------------- CM - Interpretation of a Version ---------------- 12:
Command ===>

Details for version : 'PAOLOR4'.'BASE1DSG24D04'

Version ID  . . : 178
Version owner . : PAOLOR4          Version name : BASE1DSG24D04
Created by  . . : PAOLOR4          Created  . . : 2007-03-15-12.29.00.000000
Last altered by : PAOLOR4          Last altered : 2007-03-15-12.32.35.802006
Protected . . . : YES              Type . . . . : Base
Location  . . . :                  Scope ID . . : 26
Mixed . . . . . : N                Server . . . : DB8A
Date format . . : ISO              Time format  : ISO
DB2V8NF . . . . :                  Encoding Scheme : E

GEN SSID  . . . : DB8A             GEN REL  . . . . . : 810
DECP REL  . . . : 810              DECIMAL POINT  . . : .
DECP SSID . . . : DB8A             EBCDIC CCSID . . . : 37
EBCDICM CCSID . : -2               EBCDICG CCSID  . . : -2
ASCIIS CCSID  . : 437              ASCIIM CCSID . . . : -2
ASCIIG CCSID  . : -2               UNICODES CCSID . . : 367
UNICODEM CCSID. : 1208             UNICODEG CCSID . . : 1200

Comment . . . . :
BASE VER1 DB DSNG24D04 ALL - DB TS TBS
```

Figure 6-2 CM Interpreting of a Version panel (ADB2C41I)

To see the version scope, issue the **SO** command on the CM Version Scopes panel. See Figure 6-3.

```
DB2 ADMIN -------------- CM - Version Scopes ---------------- Row 1 to 1 of 1
Command ===>                                                Scroll ===> PAGE

Line commands:
 VE - Versions  SO - Version scope objects  GV - Generate new base version
 INS - Insert  U - Update  DEL - Delete  I - Details on version scope

Sel          ID Owner    Name                Comment
              * *        *                   *
--- ----------- -------- ----------------- ---------------------------------
 SO          26 PAOLOR4  BASE1DSG24D04      BASE VER1 DB DSG224D04 ALL
****************************** END OF DB2 DATA ******************************
```

Figure 6-3 CM Version Scope panel (ADB2C42)

Archived

Then issue the **VS** command line on the CM Version panel. See Figure 6-4.

```
DB2 Admin ----------- CM - Version Scope Objects ---------
Command ===> SAVE

Version scope objects for scope "PAOLOR4"."BASE1DSG24D04"
Commands: SAVE
Line commands:
  I - Insert  D - Delete  R - Repeat

Sel T  Qual     Name               Oper.
    *  *        *                  *
--- -- -------> -----------------> ------

    DB          DSG24D04
****************************** END OF DB2 DATA *********
```

Figure 6-4 CM - Version Scope Objects (ADB2C4O)

Now that the base version has been created, you can make the changes required in the database objects.

6.1.2 Classifying changes performed using DB2 Admin

When using DB2 Admin, you should consider the type of change you want to perform.

- DB2 Alter statements are done using the DB2 Admin AL function (Alter table or materialized table).
- DB2 Drop or Create of a table or changing the data type of a column is done using DB2 Admin ALC function (Alter table columns with DB2 Admin ALTER).

Archived

Getting started with either the AL or ALC command requires you to display the object you want to modify from the DB2 Administration Menu 7.2.0 panel (ADB2) and specify **1** to review the DB2 system catalog and press Enter. See Figure 6-5.

```
DB2 Admin -------------- DB2 Administration Menu 7.2.0 ------------------ 14:00
Option ===> 1

   1 - DB2 system catalog                                DB2 System: DSN8
   2 - Execute SQL statements                            DB2 SQL ID: LUCYK
   3 - DB2 performance queries                           Userid    : LUCYK
   4 - Change current SQL ID                             DB2 Rel   : 815
   5 - Utility generation using LISTDEFs and TEMPLATEs
   P - Change DB2 Admin parameters
  DD - Distributed DB2 systems
   E - Explain
   Z - DB2 system administration
  SM - Space management functions
   W - Manage work statement lists
   X - Exit DB2 Admin
  CC - DB2 catalog copy version maintenance
  CM - Change management
                                                         More:     +
Interface to other DB2 products and offerings:
   I  DB2I

   C  DB2 Object Comparison Tool
```

Figure 6-5 DB2 Administration Menu 7.2.0 (ADB2)

The catalog holds all of the databases, table space, tables, and other objects defined on the system. This panel allows you to see the relationships of the various objects to each other and details of each object.

Archived

From the System Catalog panel (ADB21) you should specify as much as you know in terms of the name, owner, grantor, grantee, or database/ collection/schema of the object. However, these fields accept a partial name and can find a group of objects using the wild card character '%'. To see all of the tables associated with the database **DSG24D04**, you would enter only **GLWt%** for the name and **DSG24D04** for the D/L/H fields and press Enter. See Figure 6-6.

```
DB2 Admin --------------------- DB8A System Catalog --------------------- 17:0
Option ===> t

Object options:                                          DB2 System: DB8A
  AO - Authorization options                             DB2 SQL ID: PAOLOR4
   G - Storage groups                P - Plans
   D - Databases                     L - Collections
   S - Table spaces                  K - Packages
   T - Tables, views, and aliases    M - DBRMs
   V - Views                         H - Schemas
   A - Aliases                       E - User defined data types
   Y - Synonyms                      F - Functions
   X - Indexes                       O - Stored procedures
   C - Columns                       J - Triggers
   N - Constraints                   Q - Sequences
  DS - Database structures         DSP - DS with plans and packages

Enter standard selection criteria (Using a LIKE operator, criteria not saved):
Name     ===> glwt%              >  Grantor  ===>          >
Owner    ===>          >           Grantee  ===>          >
In D/L/H ===> DSG24d04           >  Switch Catalog Copy  ===> N   (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
Column   ===>                    >  Operator ===>        Value ===>
```

Figure 6-6 System Catalog (ADB21)

Archived

To perform an alter on table GLWTEMP, you enter **AL** or **ALC** next to the table name in the blank column labeled Sel and press Enter. See Figure 6-7.

```
DB2 Admin --------------- DB8A Tables, Views, and Aliases -- Row 1 to 17 of 17
Command ===>                                                  Scroll ===> PAGE

Commands: GRANT  MIG
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
 V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
 ? - Show all line commands

Sel   Name                Owner    T DB Name  TS Name    Cols        Rows Checks
      *                   *        * *        *            *           *      *
----- ------------------ -------- - -------- -------- ------ ----------- ------
      GLWTVRN             DSG24D04 T DSG24D04 GLWS001      15          -1      0
      GLWTLNG             DSG24D04 T DSG24D04 GLWS002       2          -1      0
      GLWTDPT             DSG24D04 T DSG24D04 GLWSDPT      10          -1      0
      GLWTEPA             DSG24D04 T DSG24D04 GLWSEPA      10          -1      0
AL    GLWTEMP             DSG24D04 T DSG24D04 GLWSEMP      19          -1      0
      GLWTACT             DSG24D04 T DSG24D04 GLWS001       3          -1      0
      GLWTSPL             DSG24D04 T DSG24D04 GLWSSPL       6          -1      0
      GLWTPRJ             DSG24D04 T DSG24D04 GLWSPRJ      13          -1      0
      GLWTPJA             DSG24D04 T DSG24D04 GLWSPJA      10          -1      0
      GLWTPNG             DSG24D04 T DSG24D04 GLWS002       2          -1      0
      GLWTENG             DSG24D04 T DSG24D04 GLWS002       2          -1      0
      GLWTDNG             DSG24D04 T DSG24D04 GLWS002       2          -1      0
      GLWTSFN             DSG24D04 T DSG24D04 GLWS001       3          -1      0
      GLWTPGW             DSG24D04 T DSG24D04 GLWS001       3          -1      0
      GLWTLNM             DSG24D04 T DSG24D04 GLWS001       2          -1      0
      GLWTLCN             DSG24D04 T DSG24D04 GLWS001       2          -1      0
      GLWTJBS             DSG24D04 T DSG24D04 GLWS001       4          -1      0
****************************** END OF DB2 DATA ******************************
```

Figure 6-7 Tables, Views, and Aliases panel (ADB21T)

Archived

6.1.3 Altering database objects using the AL command

All of the items shown on the Alter Table panel (ADB21TA) can be altered. In this example we add a column. You should enter / by the ADD column and press Enter. See Figure 6-8.

```
DB2 ADMIN ----------------------- DB8A Alter Table ---------------------- 17:2
Command ===>

 Table owner ===> DSG24D04 >
 Table name  ===> GLWTEMP            >

   AUDIT              ===> NONE      (None, Changes, or All)
   DATA CAPTURE       ===> NONE      (None/Changes)
   VALIDPROC          ===> NULL      (NULL/Program name)
   RESTRICT ON DROP   ===> NO        (Yes/No)
   VOLATILE           ===> NO        (Yes/No)

 ALTER TABLE with any of the above changes OR select one of the options below


 / ADD column                      ADD MATERIALIZED QUERY
   PRIMARY KEY                     DROP MATERIALIZED QUERY
   DROP PRIMARY KEY                REFRESH MATERIALIZED TABLE
   FOREIGN KEY                     ADD PARTITIONING KEY
   DROP FOREIGN KEY                ADD/ALTER PART TABLE
   ADD CHECK constraint
   DROP CHECK constraint
   ADD UNIQUE constraint
   DROP UNIQUE constraint
```

Figure 6-8 Alter Table panel (ADB21TA

Archived

Next you fill in the Column name as **ADDCOL1**, Column type as **Char**, Column length as **12.** Allow Nulls as **yes**, W. DEFAULT as **No**. Then press Enter. See Figure 6-9.

```
DB2 ADMIN ----------------------- DB8A Alter Table ---------------------- 18:0
Command ===>

 ALTER TABLE
 Table owner ===> DSG24D04 >
 Table name  ===> GLWTEMP            >
 ADD
 Column name ===> ADDCOL1            >
 Column type ===> CHAR               Built-in only
 Column leng ===> 12                   "   "   "
 Precision   ===>                      "   "   "       (opt, w/FLOAT, DECIMAL)
 Scale       ===>                      "   "   "       (opt, w/DECIMAL)
 Schema name ===>          >         User-defined only
 Data type   ===>                    >    "     "     "

 Allow Nulls ===> yes  (Yes-nullable/No-NOT NULL)
 FOR ? DATA  ===>      (B-Bit, S-SBCS, M-Mixed, blank-N/A)
 W. DEFAULT  ===> no   (Yes, No, L (SECLABEL) or enter value below)
 Def. value  ===>

 GENERATED   ===>    (A-ALWAYS, D-DFLT, I-ALWAYS AS IDENT, J-DFLT AS IDENT)
 fieldproc
 Name        ===>         (optional)
 Parm        ===>                                                  >
```

Figure 6-9 Alter Table panel (ADB21TAB)

You get prompted to use CM or not. In this example we said **yes**. See Figure 6-10.

```
EsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
e DB2 ADMIN ----------- DB8A Change Management Prompt ----------- 19:37 e
e                                                                      e
e Change Management is optional for SQLID: PAOLOR4                     e
e                                                                      e
e Do you wish to use Change Management for the statement: yes (Yes/No)  e
e                                                                      e
e                                                        More:     +   e
e Statement that is about to be executed (first 28 lines)              e
e ALTER TABLE "DSG24D04"."GLWTEMP"                                     e
e   ADD "ADDCOL1" CHAR (12)                                            e
e                                                                      e
e                                                                      e
e  F1=HELP       F2=SPLIT      F3=END        F4=RETURN     F5=RFIND     e
e  F6=RCHANGE    F7=UP         F8=DOWN       F9=SWAP      F10=LEFT      e
DsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

Figure 6-10 Change Management Prompt (ADB2CMPR)

You are prompted and get this CM - Register Change panel (ADB2CMRG). There are two possible outcomes from this panel:

- If there are NO pending changes for this object, you can choose one of these options:
  - **C** - Cancel
  - **I** - Register and run as an immediate change
  - **N** - Register as a normal change
- If there were pending changes for this same object, you can choose one of these options:
  - **C** - Cancel
  - **E** - Register and run as an emergency change
  - **N** - Register as a normal change, pending changes become prereqs
  - **S** - Register as a normal change, supersede pending changes
  - **D** - Display pending changes to the same object(s) display the pending changes,

In this example there are pending changes. We decided to first display the pending changes supplying **D** on the command line. See Figure 6-11.

```
 DB2 ADMIN ------------------ DB8A CM - Register Change ------------------ 19
 Option ===> d

  C - Cancel
  E - Register and run as an emergency change
  N - Register as a normal change, pending changes become prereqs
  S - Register as a normal change, supersede pending changes
  D - Display pending changes to the same object(s)

 For option E enter the following information for the change:
  Owner  . . . . .          >  (Optional, default is PAOLOR4)
  Name . . . . . .                  >
  Comment  . . . .
                                                                   More:
 Statement that is about to be executed (first 28 lines)
 ALTER TABLE "DSG24DO4"."GLWTEMP"
 ADD "ADDCOL1" CHAR (12)





 EsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
 e There are pending changes related to the objects you are modifying.  e
 e Use the "Display pending changes" option to see the pending changes. e
 DssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM ANGE
```

Figure 6-11 CM - Register Change (ADB2CMRG)

Pending changes are displayed and you can check with the owner of the changes (**PAOLOR3**) to see if this new alter change can supersede them or not. See Figure 6-12.

```
DB2 ADMIN ------------------ DB8A CM - Pending Changes ------- Row 1 to 2 of 2
Command ===>                                                  Scroll ===> PAGE

Line commands:
 CH - Change  I - Interpret

Sel Owner    Name                Statement
    *        *                   *
--- -------- ------------------ ---------------------------------------------->
    PAOLOR3  RECOVERY TEST        ADMIN ALTER TABLE "DSG24D04"."GLWTVRN"    ALT
    PAOLOR3  RECOVERY TEST       ADMIN ALTER TABLE "DSG24D04"."GLWTVRN"    RENA
****************************** END OF DB2 DATA ******************************
```

Figure 6-12 CM - Pending Changes (ADB2CMPC)

Paolor3 (owner) confirms that the pending changes can be superseded by our new alter change. Hence you can press PF3 to go back to the previous panel CM - Register Change (ADB2CMRG) and enter **S** to Register as a normal change, supersede pending changes, and press Enter.

Next there is a prompt in the CM - Register Options panel (ADB2CRO) to name the change and add comments to help label this change. The field Change Type is for a CHANGE (or normal change). We supply the Name as **ADCOL1GWLTEMP** and **AL ADDCOL1 CHAR 12 GWLTEMP** for the comment. The cursor can be placed below the CONTINUE word to cause this change to be registered, or the word CONTINUE can also be entered on the command line. See Figure 6-13.

```
DB2 ADMIN  ------------------ CM - Register Options -------------------- 20:08
Option ===>

Commands: CONTINUE                                    DB2 System: DB8A
                                                      DB2 SQL ID: PAOLOR4
Specify the following values to register a change:

Owner . . . . . . . PAOLOR4  >  (Optional, default is PAOLOR4)
Name  . . . . . . . ADcol1gwltemp                                         >
Comment . . . . . . al addcol1 char 12 gwltemp                            >
Change Type . . . . : CHANGE   (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                     Owner         Name
Ignore  . . . . . . .         >                          >  (? to look up)
Mask  . . . . . . . .         >                          >  (? to look up)
Delta Version . . . .         >                          >  (? to look up)
```

Figure 6-13 CM - Register Options (ADB2CRO)

The message, `Register Successful`, shows up if registration is good. See Figure 6-14.

```
DB2 ADMIN ---------------- CM - Register Options-------------------     20:21
Option ===>
Register Successful
Commands: CONTINUE                                        DB2 System: DB8A
                                                          DB2 SQL ID: PAOLOR4
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR4  >  (Optional, default is PAOLOR4)
Name  . . . . . . . . ADCOL1GWLTEMP                                        >
Comment . . . . . . . AL ADDCOL1 CHAR 12 GWLTEMP                           >
Change Type . . . . . : CHANGE   (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                      Owner         Name
Ignore  . . . . . . .         >                        >  (? to look up)
Mask  . . . . . . . .         >                        >  (? to look up)
Delta Version . . . . PAOLOR4  >    ADCOL1GWLTEMP     >  (? to look up)
```

Figure 6-14 CM - Register Options completed successfully

You can see the change by pressing PF3 a few times to get back to the DB2 Administration Menu 7.2.0 panel (ADB2) and choosing **CM** on the command line, then selecting **1** to manage changes on the Change Management (CM) panel (ADB2), then selecting **1** to displ.ay changes and supplying the owner name on the CM - Manage Changes panel (ADB2C1) to see this new change. (An alternative way to go directly to the Change Management Main Menu is by using the primary command **CMM**.) See Figure 6-15.

```
DB2 ADMIN ---------------- CM - Changes -------------------- Row 1 to 1 of 1
Command ===>                                                Scroll ===> PAGE

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
      233 PAOLOR4  ADCOL1GWLTEMP      CHANGE  DEFINED  AL ADDCOL1 CHAR 12 GW
****************************** END OF DB2 DATA ******************************
```

Figure 6-15 CM - Changes (ADB2C11)

Note: You can analyze and run the change or make more changes. We recommend that you analyze and run the change to minimize the number of pending changes for subsequent changes.

To analyze the AL changes, you enter **AN** on the line next to the change labeled `233 PAOLOR4 ADCOL1GWLTEMP`. Notice that the prior depending change (**RECOVERY TEST)** seen in the panel, Pending Changes above, `has been changed to` **CANCELED** by **Paolor3**. See Figure 6-16.

```
DB2 ADMIN ------------------ CM - Changes -------------- Row 204 to 232 of 250
Command ===>                                                Scroll ===> PAGE

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- -----------------> ------- -------- -------------------->
         149 PAOLOR3  RECOVERY TEST       CHANGE  CANCELED CHANGE CREATED FOR RE
         .
         .
         .
AN       233 PAOLOR4  ADCOL1GWLTEMP       CHANGE  DEFINED AL ADDCOL1 CHAR 12 GW
```

Figure 6-16 Changes (ADB2C11)

Archived

We are prompted to enter fields in the panel, Generate Analyze Job. In this case, we automatically generate the base version by entering **a** for auto, Validate WSL = **yes**, Build the JCL for running WSL = **yes**, and for most of the other values, we enter **n** for no. See Figure 6-17.

```
DB2 Admin ----------------- Generate Analyze Job  ----------------------- 19:24
Command ===>
 Specify the following for Analyze:

 Base version method . . . . . a (Auto, User, or Existing)

 Change reporting options  . . no (Yes/No)

 Required data set information:
   PDS for WSL . . . . . . . DB8A.ANALYZE.WSL1
   PDS for batch jobs  . . . DB8A.ANALYZE.JCL1
   Prefix for data sets  . .  paolor4

 Options:
   Run SQLID . . . . . . . .          (Blank, a SQLID, or <NONE>)
   Validate WSL  . . . . . . yes      (Yes/No)
   Use utility options . . . NO       (Yes/No)
   Generate templates  . . . NO       (Yes/No)
   Build JCL to run WSL  . . yes      (Yes/No)
   Generate a recover change n        (Yes/No)
     Data to recover . . . .          (Original or Existing)


   Optional jobs after Reload or Alter:
     Run CHECK DATA  . . . . n        (Yes/No)
     Take an image copy  . . n        (after: Reload/Alter/Both/None)
     Run REORG . . . . . . . n        (Mandatory, All relevant, None)
     Run RUNSTATS  . . . . . n        (after: Reload/Alter/Both/None)
     Run REBIND  . . . . . . n        (Yes/No)

 BP - Change batch job parameters
 TU - Specify TEMPLATE usage
 UO - Change utility options
```

Figure 6-17 Generate Analyze Job (ADB2C11A)

Archived

Next, DB2 Admin prompts you to decide the scope of the change and if a new version is to be created in the panel, Run a Change. In this case we create a new base version named NBASEGWLTEMP. See Figure 6-18.

```
DB2 ADMIN ---------------- CM - Run a Change ----------------------------------
Command ===>
Enter required field

Change . . . : PAOLOR4.ADCOL1GWLTEMP


Data set information:
  PDS for batch jobs . . . . . 'PAOLOR4.DB8A.RUN.JCL'
  Prefix for data sets . . . . PAOLOR4

Create a new base version  . . yes (Yes/No)

  Scope Information:
    Owner  . . . . . . . . . . paolor4  >           (? to look up)
    Name . . . . . . . . . . . base1DSG24D04      >  (? to look up)

  Version Information:
    Owner  . . . . . . . . . .  paolor4 >           (? to look up)
    Name . . . . . . . . . . . NBASEGWLTEMP>        (? to look up)

Runtime analyze:
  Change reporting options . . YES  (Yes/No)
```

Figure 6-18 Run a Change (ADB2CEX1)

You get prompted to run the job. Key in **sub**, and check that the job runs to **rc=0**. Then you can review the version files in CM and you now see the base version and the new base. See Figure 6-19.

```
DB2 ADMIN ----------------- CM - Versions ------------------ Row 1 to 2 of 2
Command ===>                                              Scroll ===> PAGE

Line commands:
 CH - Changes  PR - Promote  VS - Version scope  DEL - Delete  U - Update
 PT - Toggle protected status  I - Details on version

Sel          ID T Owner    Name                Comment
              * * *        *                   *
--- ----------- - -------- ----------------- ------------------------------>
            178 B PAOLOR4  BASE1DSG24D04      BASE VER1 DB DSNG24D04 ALL - DB
            201 D PAOLOR4 NBASEGWLTEMP
****************************** END OF DB2 DATA ******************************
```

Figure 6-19 Versions (ADB2C41)

Archived

Next you would want to run the change. Go to panel Changes and enter **RN** next to the change ADCOL1GWLTEMP. See Figure 6-20.

```
DB2 ADMIN ------------------ CM - Changes -------------------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> PAGE

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
rn       233 PAOLOR4  ADCOL1GWLTEMP       CHANGE  ANALYZED AL ADDCOL1 CHAR 12 GW
****************************** END OF DB2 DATA ******************************
```

Figure 6-20 Changes (ADB2C11)

You get prompted, `Run a Change`, and you input the data set name and enter **N**, in order not to create a new base version. See Figure 6-21.

```
DB2 ADMIN ---------------- CM - Run a Change ---------------------------
Command ===>


Change . . . : PAOLOR4.ADCOL1GWLTEMP


Data set information:
  PDS for batch jobs . . . . . 'PAOLOR4.PAOLOR4.ALCOL1'
  Prefix for data sets . . . . PAOLOR4

Create a new base version  . . n    (Yes/No)

  Scope Information:
    Owner  . . . . . . . . . .         >           (? to look up)
    Name . . . . . . . . . . .                   >  (? to look up)

  Version Information:
    Owner  . . . . . . . . . .         >           (? to look up)
    Name . . . . . . . . . . .                   >  (? to look up)

Runtime analyze:
  Change reporting options . . YES  (Yes/No)
```

Figure 6-21 Run a Change (ADB2CEX1)

Archived

Once you press Enter, you are prompted to decide the reporting options in the panel, Specify Compare Reporting Options. Once you have entered them, then you again press Enter and get prompted to run the job. When you enter **sub,** the job runs. Going back to the Change panel, you can display the change and see that **ADCOL1GWLTEMP** has now changed to a Status of **COMPLETE**. See Figure 6-22.

```
DB2 ADMIN ------------------ CM - Changes -------------------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> PAGE

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
         233 PAOLOR4  ADCOL1GWLTEMP      CHANGE  COMPLETE AL ADDCOL1 CHAR 12 GW
```

Figure 6-22 Changes (ADB2C11)

6.1.4 Using the ALC command to implement table changes

An alternative way to make changes to the database is by using the alter column (ALC). In this example we add a foreign key to a table.

As before, you go to the DB2 System Catalog option **1** on the DB2 Administration Menu 7.2.0 panel and supply the table name you would like to alter or a generic name using the wild character of '%' in the name field. For this example, we are choosing to alter the GLWTDPT table. We enter **ALC** next to the table name from the Tables, Views, and Aliases panel (ADB21T). See Figure 6-23.

```
DB2 ADMIN -------------- DB8A Tables, Views, and Aliases -- Row 1 to 17 of 17
Command ===>                                                  Scroll ===> PAGE

Commands: GRANT  MIG
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
 V - Views  T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
 ? - Show all line commands

Sel   Name                Owner    T DB Name  TS Name    Cols        Rows Checks
      *                   *        * *        *             *           *      *
----- ------------------ -------- - -------- -------- ------ ----------- ------
alc   GLWTDPT             DSG24D04 T DSG24D04 GLWSDPT      10          -1      0
****************************** END OF DB2 DATA ******************************
```

Figure 6-23 Tables, Views, and Aliases (ADB21T)

Once again you are prompted to use CM or not. We enter **YES**. See Figure 6-24.

```
EsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
e DB2 ADMIN ----------- DB8A Change Management Prompt ----------- 18:15 e
e                                                                      e
e Change Management is optional for SQLID:  PAOLOR4                    e
e                                                                      e
e Do you wish to use Change Management for this function: yes (Yes/No)  e
e                                                                      e
e  F1=HELP       F2=SPLIT      F3=END        F4=RETURN     F5=RFIND     e
e  F6=RCHANGE    F7=UP         F8=DOWN       F9=SWAP      F10=LEFT      e
DssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

Figure 6-24 Change Management Prompt (ADB2CMRO)

You alter a primary key for this table by updating DEPT_NO by entering a **U** in the Sel column and pressing Enter. See Figure 6-25.

```
DB2 ADMIN ------------------- DB8A ALTER Table ------------- Row 1 to 10 of 10
Command ===>                                                 Scroll ===> CSR

New owner ===> DSG24D04 >                      Old owner : DSG24D04
New name  ===> GLWTDPT           >             Old name  : GLWTDPT
Volatile  ===> NO        Action    : NONE      New DB ===> DSG24D04
Rows/Page    : 40.554    Partitions: 0         New TS ===> GLWSDPT
Commands :  CONTINUE  PRIMKEY  ADD  REL
Line commands :
 I - Insert  U - Update  D - Delete  R - Repeat  LAB - Label  COM - Comment
 M - Move    A - After   B - Before  X - Index   RES - Reset update
                                                          Old Operation
Select Column Name        Col No Col Type Length  Scale Null D Col No Type
       *                   * *            *      * *    *      * *
------ ----------------> ------ -------- ------ ------ ---- - ------ ---------

  u    DEPT_NO                1 INTEGER       4      0 N    Y      1
       DEPT_NAME              2 CHAR         22      0 N    Y      2
       MGR_NO                 3 INTEGER       4      0 Y    Y      3
       ADMR_DEPT              4 INTEGER       4      0 N    Y      4
       LOCATION               5 CHAR         20      0 N    Y      5
       EMPLOYEE_CNT           6 SMALLINT      2      0 N    Y      6
       CREATED_TS             7 TIMESTMP     10      0 N    Y      7
       CREATED_BY             8 CHAR          8      0 N    Y      8
       UPDATED_TS             9 TIMESTMP     10      0 N    Y      9
       UPDATED_BY            10 CHAR          8      0 N    Y     10
****************************** END OF DB2 DATA ******************************

 EsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
 e Related objects INDEXES=Y VIEWS=Y PLANS=N RIs=N PACKAGES=Y TRIGGERS=N e
 DssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

Figure 6-25 Alter Table (ADB27C)

This brings up a new line on the Alter Table panel indicating UP should be used to update the primary key. See Figure 6-26.

```
DB2 ADMIN ------------------- DB8A ALTER Table ------------- Row 1 to 10 of 10
Command ===>                                                  Scroll ===> CSR
Use UP,DP to upd,del
New owner ===> DSG24DO4 >                      Old owner : DSG24DO4
New name  ===> GLWTDPT            >            Old name  : GLWTDPT
Volatile  ===> NO        Action    : NONE      New DB ===> DSG24DO4
Rows/Page    : 40.554    Partitions: 0         New TS ===> GLWSDPT
Commands :  CONTINUE  PRIMKEY  ADD  REL
Line commands : UP - Update Primary Key
 I - Insert  U - Update  D - Delete  R - Repeat  LAB - Label  COM - Comment
 M - Move    A - After   B - Before  X - Index   RES - Reset update
                                                            Old Operation
Select Column Name        Col No Col Type Length  Scale Null D Col No Type
       *                     * *            *      * *    *     * *
------ -----------------> ------ -------- ------ ------ ---- - ------ ---------

U      DEPT_NO                1 INTEGER       4      0 N    Y      1
       DEPT_NAME              2 CHAR         22      0 N    Y      2
       MGR_NO                 3 INTEGER       4      0 Y    Y      3
       ADMR_DEPT              4 INTEGER       4      0 N    Y      4
       LOCATION               5 CHAR         20      0 N    Y      5
       EMPLOYEE_CNT           6 SMALLINT      2      0 N    Y      6
       CREATED_TS             7 TIMESTMP     10      0 N    Y      7
       CREATED_BY             8 CHAR          8      0 N    Y      8
       UPDATED_TS             9 TIMESTMP     10      0 N    Y      9
       UPDATED_BY            10 CHAR          8      0 N    Y     10
****************************** END OF DB2 DATA ******************************
```

Figure 6-26 Alter Table (ADB27C)

Archived

Next we enter **UP** in the column Select next to DEPT_NO to update this primary key. This takes you to the panel, Create Table Column Number 1 to alter this primary key. See Figure 6-27.

```
DB2 ADMIN ------------- DB8A Create Table Column Number 1   ------------- 21:22
Command ===>

 DB2 Admin ALTER                        Owner ===> DSG24D04
                                        Name  ===> GLWTDPT
 Column name ===> DEPT_NO               (column number 1)

 Data type   ===> INTEGER            (CHAR,DECIMAL,INTEGER,SMALLINT,etc)
 Data length ===>
 Precision   ===>                    (used only w/FLOAT and DECIMAL)
 Scale       ===>                    (used only w/DECIMAL)
 Schema name ===>                    (User-defined only)
 Type name   ===>

 Allow Nulls ===> NO   (Yes-nullable/No-NOT NULL)
 FOR ? DATA  ===>      (B-Bit, S-SBCS, M-Mixed, blank-N/A)
 w/DEFAULT   ===> YES  (Yes, No, L (SECLABEL) or enter value below)
 Def. value  ===>

 GENERATED   ===>    (A-ALWAYS, D-DFLT, I-ALWAYS AS IDENT, J-DFLT AS IDENT)
 FIELDPROC
 Program name ===>         (optional)
 Program parm ===>                                                      >
```

Figure 6-27 Create Table Column Number 1 (ADB26CTU)

Archived

You can then change the DEPT_NO to data type of decimal, precision of 4, and scale of 0. See Figure 6-28.

```
DB2 ADMIN ------------- DB8A Create Table Column Number 1   ------------- 21:22
Command ===>

DB2 Admin ALTER                          Owner ===> DSG24D04
                                         Name  ===> GLWTDPT
 Column name ===> DEPT_NO                (column number 1)

 Data type   ===> DECIMAL             (CHAR,DECIMAL,INTEGER,SMALLINT,etc)
 Data length ===>
 Precision   ===> 4                   (used only w/FLOAT and DECIMAL)
 Scale       ===> 0                   (used only w/DECIMAL)
 Schema name ===>                     (User-defined only)
 Type name   ===>

 Allow Nulls ===> NO   (Yes-nullable/No-NOT NULL)
 FOR ? DATA  ===>      (B-Bit, S-SBCS, M-Mixed, blank-N/A)
 w/DEFAULT   ===> YES  (Yes, No, L (SECLABEL) or enter value below)
 Def. value  ===>

 GENERATED   ===>    (A-ALWAYS, D-DFLT, I-ALWAYS AS IDENT, J-DFLT AS IDENT)
 FIELDPROC
 Program name ===>          (optional)
 Program parm ===>                                                      >
```

Figure 6-28 Create Table Column Number 1 (ADB25CTU)

Once you press Enter, you are taken to a warning panel, DB2 Admin ALTER Column Truncation. This allows you to specify what DB2 Admin should do if conversion of the data prevents the data to be reloaded into the field. We entered **Z** to set the field to zero for the action. See Figure 6-29.

```
DB2 ADMIN ------------ DB8A DB2 Admin ALTER Column Truncation ----------- 21:40
Option ===>

The new column data type may be invalid or truncated at reload time.
Please specify action below.

Old definition:                          New definition:
  Column name . : DEPT_NO                  Column name . : DEPT_NO
  Data type . . : INTEGER                  Data type . . : DECIMAL
  Length  . . . :      4                   Length  . . . :      4
  Scale . . . . :      0                   Scale . . . . :      0

Specify what DB2 Admin should do if field is too large:
  Action  . . . : Z  D - Let DB2 LOAD utility put it on DISCARDS file
                     Z - Set the field to zero (or blank for char columns)
                     T - Truncate field value
                     C - Cancel update
```

Figure 6-29 DB2 Admin ALTER Column Truncation (ADB27CT)

Pressing Enter takes you back to the ALTER Table panel (ADB27C) showing that the DEPT_NO column has been updated and Action is now DROP. Notice that **ADDFK** command was added to the panel. See Figure 6-30.

```
 DB ADMIN ------------------- DB8A ALTER Table ------------- Row 1 to 10 of 10
 Command ===>                                                 Scroll ===> CSR

 New owner ===> DSG24D04 >                      Old owner : DSG24D04
 New name  ===> GLWTDPT            >            Old name  : GLWTDPT
 Volatile  ===> NO        Action    : DROP      New DB ===> DSG24D04
 Rows/Page    : 39.384    Partitions: 0         New TS ===> GLWSDPT
 Commands :  CONTINUE  PRIMKEY  ADD  REL           ADDFK
 Line commands :
  I - Insert  U - Update  D - Delete  R - Repeat  LAB - Label  COM - Comment
  M - Move    A - After   B - Before  X - Index   RES - Reset update
                                                                   Old
Operation
 Select Column Name        Col No Col Type Length  Scale Null D Col  No Type
        *                      * *            *      * *    *      * *
 ------ ----------------> ------ -------- ------ ------ ---- - ------ ---------

 *      DEPT_NO                 1 DECIMAL      4      0 N    Y      1 UPDATE
        DEPT_NAME               2 CHAR        22      0 N    Y      2
        MGR_NO                  3 INTEGER      4      0 Y    Y      3
        ADMR_DEPT               4 INTEGER      4      0 N    Y      4
        LOCATION                5 CHAR        20      0 N    Y      5
        EMPLOYEE_CNT            6 SMALLINT     2      0 N    Y      6
        CREATED_TS              7 TIMESTMP    10      0 N    Y      7
        CREATED_BY              8 CHAR         8      0 N    Y      8
        UPDATED_TS              9 TIMESTMP    10      0 N    Y      9
        UPDATED_BY             10 CHAR         8      0 N    Y     10
 ****************************** END OF DB2 DATA******************************
```

Figure 6-30 Alter Table (ADB27C)

Entering **Continue** on the command line, or placing the cursor below the word and pressing Enter, checks the RI relationship of the GWLTDPT table repeatedly and adds the related tables to the list. The column change performed on DEPT_NO is propagated to the corresponding columns of the RI related tables and you are prompted on the Alter Related:DSG24D04.GLWTDPT panel to alter the unique index table that has DEPT_NO.

You can enter **a** beside the index table GLWXDPT1, which has the DEPT_NO column in it. Notice that you can use the **S** line command to get information about any of the objects listed if you do not know which object you should change. In this case the only related table for DEPT_NO is GLWXDPT1. See Figure 6-31.

```
DB2 Admin ------- DB8A Alter Related : DSG24DO4.GLWTDPT ------ Row 1 to 4 of 4
Command ===>                                                  Scroll ===> PAGE
Primary index required.
Commands: ADDINDEX   ADDFK
Line commands:  A - Alter object  S - Show object

                                                 Dep.
S Type        Creator  Name                    Views Rel Name Note
  *           *        *                           * *        *
- ----------- -------- ------------------ ------ -------- ---------------------
a INDEX       DSG24DO4 GLWXDPT1                    0
  INDEX       DSG24DO4 GLWXDPT2                    0
  TABLESPACE  DSG24DO4 GLWSDPT                     0
  VIEW        DSG24DO4 WLDEPT                      0
****************************** END OF DB2 DATA ******************************
```

Figure 6-31 Alter Tables (ADB27CA)

After the **a** was entered by GLWXDPT1 and you pressed Enter, the Redefine Index panel is displayed, showing that the unique index column DEPT_NO has the new decimal value. See Figure 6-32.

```
DB2 Admin --------------- DB8A Redefine Index ------------------ Row 1 from 10
Command ===>                                                  Scroll ===> PAGE

Commands: CONTINUE  ORIGINAL
Line commands: nnn A|D - Sequence & order  R - Remove the column
 A - Ascending  D - Descending

CREATE INDEX DSG24DO4   . GLWXDPT1          >
          ON DSG24DO4.GLWTDPT
Unique         ===> YES      Where Not Null  ===>      Cluster      ===> YES
Buffer pool    ===> BP16     Close rule      ===> YES  Copy Allowed ===> NO
Piece size     ===> 2097152  Define          ===> YES  Defer        ===>
Partitioned    ===>          Padded          ===>

Select Column Name        Col Type Length  Scale Null ColSeq Order OldSeq Order
       *                  *             *      * *         * *          * *
------ ----------------- -------- ------ ------ ---- ------ ----- ------ -----
       DEPT_NO            DECIMAL       4      0 N         1 A          1 A
       DEPT_NAME          CHAR         22      0 N
       MGR_NO             INTEGER       4      0 Y
       ADMR_DEPT          INTEGER       4      0 N
       LOCATION           CHAR         20      0 N
       EMPLOYEE_CNT       SMALLINT      2      0 N
       CREATED_TS         TIMESTMP     10      0 N
       CREATED_BY         CHAR          8      0 N
       UPDATED_TS         TIMESTMP     10      0 N
       UPDATED_BY         CHAR          8      0 N
****************************** END OF DB2 DATA ******************************
```

Figure 6-32 Redefine Index (ADB21XAR)

To redefine the existing index, enter CONTINUE on the command line or place the cursor under the word CONTINUE and press Enter.

You are then prompted to confirm this change by seeing Clustering Index Confirmation panel. To continue, you would enter **1** to select Continue specification as a clustering index. See Figure 6-33.

```
DB2 ADMIN -- DB8A Clustering Index Confirmation ------------------- 13:50

A clustering index already exists for the table.
Specify whether to continue or return.



Select a choice
1 1.  Continue specification as a clustering index.
  2.  Continue, but remove clustering index clause.
  3.  Display the current clustering index.
  4.  Return
```

Figure 6-33 Clustering Index Confirmation (ADB2CONF)

Once you press Enter, the Redefine Index - Space panel is displayed. Here you can specify CONTINUE or go back to the original index values. In this example we continue by placing the cursor below the word CONTINUE. See Figure 6-34.

```
DB2 ADMIN --------------- DB8A Redefine Index - Space -------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> PAGE

Commands: CONTINUE  ORIGINAL
Line commands: O - Original data  C - Clear data

CREATE INDEX DSG24D04.GLWXDPT1
         ON DSG24D04.GLWTDPT
Sel   Part        Pqty   Sqty FreePg %Free Erase ST VCAT     Stogroup GBPCache
        *           *      *      *     * *      * *        *        *
--- ------ ----------- ------ ------ ----- ----- -- -------- -------> --------

        0          52     52      0    10 NO     I DB8AU    GLWG01   CHANGED
****************************** END OF DB2 DATA ******************************
```

Figure 6-34 Redefine Index - Space (ADB21XAS)

Then panel ISREDDE2 is displayed to show you the Create statements to be generated. At this point you can press PF3 to go back. See Figure 6-35.

```
ISREDDE2   SYS07080.T135826.RA000.PAOLOR4.R0111656         Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
****** **************************** Top of Data ******************************
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001   CREATE UNIQUE INDEX
000002   "DSG24D04"."GLWXDPT1"
000003   ON
000004   "DSG24D04"."GLWTDPT"
000005     ( "DEPT_NO" ASC)
000006     USING STOGROUP GLWG01
000007           PRIQTY 52
000008           SECQTY 52
000009     FREEPAGE 0
000010     PCTFREE  10
000011     GBPCACHE CHANGED
000012     DEFINE YES
000013     CLUSTER
000014     BUFFERPOOL BP16
000015     CLOSE YES
000016     COPY  NO
000017     PIECESIZE 2097152 K
****** ***************************** Bottom of Data ****************************
```

Figure 6-35 ISREDDE2

After you press PF3 to go back to the Alter Tables panel, you can enter **ALTER** on the command line or place the cursor below the word. Notice that both table and index have been added. See Figure 6-36.

```
DB2 ADMIN ------------------------ DB8A Alter Tables ---------- Row 1 to 2 of 2
Command ===>                                                   Scroll ===> CSR

Commands:
  ALTER - generate jobs   ADDFK - Add Foreign Key-affected tables
Line commands:
  A - Alter Object  D - Delete  S - Select Object  REL - Alter related
 FK - Add Foreign Key-affected tables  E - Edit DDL  CD - Convert to Drop
 RS - Reset RI-FK flags

To add a table, provide an owner and name below and hit Enter
   Owner   ===>         >            (Creator)
   Name    ===>                   >  (Table Name.  ? to look up)
                                                     RI RI  FK  Chg Object
Sel Creator  Name               T DB Name  TS Name    Rels Add Add Rqd Action
    *        *                  * *        *            * *   *    *  *
--- -------- ------------------ - -------- -------- ------ --- --- --- -------
    DSG24D04 GLWXDPT1           X                          NA  NA      DROP
*EL DSG24D04 GLWTDPT            T DSG24D04 GLWSDPT       0 NA  NA      DROP
****************************** END OF DB2 DATA ******************************
```

Figure 6-36 Alter Tables (ADB27CA)

Archived

6.1.5 Registering a change through ALC

You now get prompted to register this ALC change in the panel, Register Options. Notice that this change is a normal change with change type = Change. See Figure 6-37.

```
DB2 AMIN  -------------------- CM - Register Options -------------------- 22:01
Option ===>

Commands: CONTINUE                                        DB2 System: DB8A
                                                          DB2 SQL ID: PAOLOR4
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR4  >  (Optional, default is PAOLOR4)
Name  . . . . . . . . ALCPRIMKEY1 >
Comment . . . . . . . ALC CHANGE PRIMKEY DEPT_NO                            >
Change Type . . . . : CHANGE   (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                      Owner         Name
Ignore  . . . . . . .          >                          >  (? to look up)
Mask  . . . . . . . .          >                          >  (? to look up)
Delta Version . . . .          >                          >  (? to look up)
```

Figure 6-37 Register Options (ADB2CRO)

By pressing Enter, notice that the Owner and Name fields get filled in automatically for the Delta Version. Now you can place the cursor under the word CONTINUE, or enter **CONTINUE** on the command line, and then you see the message `Register Successful` displayed. See Figure 6-38.

```
DB2 ADMIN -------------------- CM - Register Options -------------------- 23:21
Option ===>
Register Successful
Commands: CONTINUE                                        DB2 System: DB8A
                                                          DB2 SQL ID: PAOLOR4
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR4  >  (Optional, default is PAOLOR4)
Name  . . . . . . . . ALCPRIMKEY1                                           >
Comment . . . . . . . ALC CHANGE PRIMKEY DEPT_NO                            >
Change Type . . . . : CHANGE   (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                      Owner         Name
Ignore  . . . . . . .          >                          >  (? to look up)
Mask  . . . . . . . .          >                          >  (? to look up)
Delta Version . . . . PAOLOR4  >    ALCPRIMKEY1           >  (? to look up)
```

Figure 6-38 Register Options (ADB2CRO)

6.1.6 ANALYZE: Analyzing a change

You can go to CM on the Main panel and then display the changes to be reviewed on the Changes panel. From this panel you can enter **AN** next to the ALC change to analyze the change. See Figure 6-39.

```
DB2 ADMIN ------------------ CM - Changes -------------------- Row 1 to 2 of 2
Command ===>                                                  Scroll ===> PAGE

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
         233 PAOLOR4  ADCOL1GWLTEMP       CHANGE  COMPLETE AL ADDCOL1 CHAR 12 GW
an       290 PAOLOR4  ALCPRIMKEY1         CHANGE  DEFINED  ALC CHANGE PRIMKEY DE
****************************** END OF DB2 DATA ******************************
```

Figure 6-39 Changes (ADB2C11)

Once you press Enter, you are prompted to fill in values required for the ANALYZE job. In this case we specify the option to create a recover change and use original data.

Tip: It is a good idea to create a recover change when making changes. A recover change is a change that lets you back out a change that has been completed.

Archived

See Figure 6-40 for the Generate Analyze Job panel.

```
DB2 Admin ----------------- Generate Analyze Job  ---------------------- 15:01
Command ===>

 Specify the following for Analyze:

 Base version method . . . . . E        (Auto, User, or Existing)

 Change reporting options  . . YES      (Yes/No)

 Required data set information:
   PDS for WSL . . . . . . . 'DB8A.ANALYZE.WSL8'
   PDS for batch jobs  . . . DB8A.ANALYZE.JCL8
   Prefix for data sets  . . PAOLOR4

 Options:
   Run SQLID . . . . . . . .          (Blank, a SQLID, or <NONE>)
   Validate WSL  . . . . . . NO       (Yes/No)
   Use utility options . . . NO       (Yes/No)
   Generate templates  . . . NO       (Yes/No)
   Build JCL to run WSL  . . YES      (Yes/No)
   Generate a recover change YES      (Yes/No)
     Data to recover . . . . O        (Original or Existing)


   Optional jobs after Reload or Alter:
     Run CHECK DATA  . . . . NO       (Yes/No)
     Take an image copy  . . N        (after: Reload/Alter/Both/None)
     Run REORG . . . . . . . N        (Mandatory, All relevant, None)
     Run RUNSTATS  . . . . . N        (after: Reload/Alter/Both/None)
     Run REBIND  . . . . . . NO       (Yes/No)

 BP - Change batch job parameters
 TU - Specify TEMPLATE usage
 UO - Change utility options
```

Figure 6-40 Generate Analyze Job (ADB2C11A)

Archived

Once you press Enter, you get prompted to name the recover job name. See Figure 6-41.

```
DB2 Admin -------------------- CM - Register Options -------------------- 15:20
Option ===>

Commands: CONTINUE                                      DB2 System: DB8A
                                                        DB2 SQL ID: PAOLOR4
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR4  >  (Optional, default is PAOLOR4)
Name  . . . . . . . . RALCPRIMKY1                                          >
Comment . . . . . . . RECCHANGE FOR CHGID290                               >
Change Type . . . . . : RECOVER  (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                      Owner         Name
Ignore  . . . . . . .         >                        >  (? to look up)
Mask  . . . . . . . .         >                        >  (? to look up)
Delta Version . . . . PAOLOR4  >    RALCPRIMKY1         >  (? to look up)
```

Figure 6-41 Setting the recovery job name

The job output from the analyze ended with **Rc=0** and shows the changes in Figure 6-42.

```
Compare table source(DSG24DO4.GLWTDPT) and target(DSG24DO4.GLWTDPT)
  Column DEPT_NO
    (D)Type changed from INTEGER to DECIMAL(4, 0)
  Tables have identical column lists
  Table DSG24DO4.GLWTDPT will be dropped
  Table will be recreated
  Table data will be converted

Compare index source(DSG24DO4.GLWXDPT1) and target(DSG24DO4.GLWXDPT1)
  Index DSG24DO4.GLWXDPT1 will be dropped by dropping the table
  Index will be recreated because the base table will be dropped and recreated

Compare index source(DSG24DO4.GLWXDPT2) and target(DSG24DO4.GLWXDPT2)
  Index DSG24DO4.GLWXDPT2 will be dropped by dropping the table
  Index will be recreated because the base table will be dropped and recreated

View DSG24DO4.WLDEPT will be recreated because it is dependent on an object
that
```

Figure 6-42 Job output

Archived

You can review the ALC changes and see that the status has changed to ANALYZED. In addition, a recover change has been created and its status is also ANALYZED. See Figure 6-43.

```
DB2 Admin ------------------ CM - Changes -------------------- Row 1 to 9 of 9
Command ===>                                                   Scroll ===> PAGE

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
         233 PAOLOR4  ADCOL1GWLTEMP       CHANGE  COMPLETE AL ADDCOL1 CHAR 12 GW
         290 PAOLOR4  ALCPRIMKEY1         CHANGE  ANALYZED ALC PRIMKEY DEPT_NO
         294 PAOLOR4  RALCPRIMKY1         RECOVER ANALYZED RECCHANGE FOR CHGID29
****************************** END OF DB2 DATA ******************************
```

Figure 6-43 Changes in ANALYZED Status

6.1.7 RUN: Running a change and creating a new version file

Once the ALC change has been analyzed, you can run it by entering **RN** on the Change panel. In this example, we specify **yes** to have a new base version created, using a scope that we created. To select the scope, your can specify your ID in the Owner field and type a **?** in the name field to find the scope name. See Figure 6-44.

```
DB2 ADMIN --------------- CM - Run a Change ------------------------------
Command ===>

Change . . . : PAOLOR4.ALCPRIMKEY1


Data set information:
  PDS for batch jobs . . . . . 'PAOLOR4.PAOLOR4.ALCOL1'
  Prefix for data sets . . . . PAOLOR4

Create a new base version  . . yes  (Yes/No)

  Scope Information:
    Owner  . . . . . . . . . . paolor4  >           (? to look up)
    Name . . . . . . . . . . . ?                  >  (? to look up)

  Version Information:
    Owner  . . . . . . . . . .          >           (? to look up)
    Name . . . . . . . . . . .                    >  (? to look up)

Runtime analyze:
  Change reporting options . . YES  (Yes/No)
```

Figure 6-44 Run a Change (ADB2CEX1)

You are prompted with the available scopes for their ID in the panel, Version Scopes. You should enter **+** on the command line by the scope you want. In this case it is **BASE1DSG24D04**. See Figure 6-45.

```
DB2 ADMIN --------------- CM - Version Scopes ---------------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> PAGE
Select by typing '+'
Line commands:
 VE - Versions  SO - Version scope objects  GV - Generate new base version
 INS - Insert  U - Update  DEL - Delete  I - Details on version scope

Sel          ID Owner    Name                Comment
              * *        *                   *
--- ----------- -------- ------------------ ----------------------------------
 +           26 PAOLOR4  BASE1DSG24D04      BASE VER1 DB DSG224D04 ALL
```

Figure 6-45 Version Scopes (ADB2C42)

Once you press Enter, you are taken back to the Run a Change panel. Notice that the scope name was filled in. Then you should enter the Version owner ID and new base version name. In this example the owner is **PAOLOR4** and the version name is **ALCPRIMKEY**. See Figure 6-46.

```
DB2 ADMIN ---------------- CM - Run a Change -------------------------
Command ===>

Change . . . : PAOLOR4.ALCPRIMKEY


Data set information:
  PDS for batch jobs . . . . . 'PAOLOR4.PAOLOR4.ALCOL1'
  Prefix for data sets . . . . PAOLOR4

Create a new base version  . .  yes (Yes/No)

  Scope Information:
    Owner  . . . . . . . . . . PAOLOR4  >            (? to look up)
    Name . . . . . . . . . . . BASE1DSG24D04      >  (? to look up)

  Version Information:
    Owner  . . . . . . . . . . PAOLOR4  >            (? to look up)
    Name . . . . . . . . . . . BASE2DSG24D04      >  (? to look up)
Runtime analyze:
  Change reporting options . . YES  (Yes/No)
```

Figure 6-46 Run a Change (ADB2CEX1)

Once the run job completes with **rc=0**, you can review both the Changes panel to verify that the changes have run successfully and also review your version file.

Archived

6.1.8 Changes in the technical design

Some changes are more technical in design, and change such items as:

- Size
- Buffer pool
- Storage group
- Close rules

Some of these changes are often unique to a system and are not migrated or promoted to another system, while other technical design changes often are migrated or promoted to many of the other systems. These might include:

- Compression
- Partitioning
- Lock size

Most of the technical changes mentioned above would be implemented using the DB2 Alter AL command.

Next we discuss an example of changing a simple table space into a partitioned table space. As before, we display a table space from the Table Spaces panel (ADB21S) and issue the command **AL** in the Sel row next to the table space we want to change. See Figure 6-47.

```
DB2 ADMIN ------------------ DSN8 Table Spaces --------------- Row 1 to 1 of 1

Commands: GRANT  MIG  DIS  STA  STO
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line commands

Select Name     DB Name   Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
       *        *            * *      * * * * *      *           *      * * *
------ -------- -------- ------ ------ - - - - - ------ ----------- ------ - -
AL     TSREDBK  DBREDBK       0 BP0    A N A N Y      1          -1     16   Y
****************************** END OF DB2 DATA ******************************
```

Figure 6-47 Table Space (ADB21S)

This takes you to the Alter Table Space panel (ADB21SA) in which you can issue the **RDEF** command to redefine (alter) the table space from a simple (or segmented) to a partitioned table space. See Figure 6-48.

```
DB2 ADMIN -------------------- DSN8 Alter Table Space -------- Row 1 to 1 of 1

Commands: RDEF
Line commands:
 D - Display Database   I - Interpret

ALTER TABLESPACE : DBREDBK.TSREDBK        (Nonpartitioned           )

Buffer pool     ===> BP0        Close rule   ===> YES Max Rows ===> 255
Lock Size       ===> ANY                              Lock Max ===> SYSTEM

         Primary     Secondary   Free Pct  Com E T S
S   Part Quantity    Quantity    Page Free prs R M T VCAT     Stogroup GBPCache
- ------ ----------- ----------- ---- ---- --- - - - -------- -------> -------

       0          48          48    0    0 YES N Y I DSN8     SYSDEFLT CHANGED
****************************** END OF DB2 DATA ******************************

Command ===> rdef                                               Scroll ===> CSR
```

Figure 6-48 Alter Table Space (ADB21SA)

You are prompted to register this change in the CM database via Change Management Prompt panel (ADB2CMRO). Choosing **Yes** on this panel registers the change and allows you to do all the Change Management functions. Next you can change the Numparts to a **2** to add two partitions and press Enter. See Figure 6-49.

```
DB2 ADMIN ----------------- DSN8 Redefine Table Space ------- Row 1 to 1 of 1

Commands: CONTINUE  ORIGINAL
Line commands: S - Split part   R - Remove part  O - Original data
               C - Clear data
CREATE TABLESPACE: TSREDBK  IN  DBREDBK

Numparts        ===> 2          Large      ===> NO    LOB      ===> NO
Define          ===> YES        DSSIZE     ===>       LOG      ===>
Member Cluster ===> NO          SEGSIZE    ===> 16    CCSID    ===> EBCDIC
Buffer Pool     ===> BP0        Close Rule ===> YES   Max Rows ===> 255
Lock Size       ===> ANY        Lock Part  ===>       Lock Max ===> SYSTEM

                                Free  Pct       E T S
S   Part  Pqty           Sqty Page Free Compr  R M T VCAT   Stogroup   BPCache
- ------ ---------- ---------- ---- ---- ----- - - - ------- ------>--------

      0        48         48    0    0 YES   N Y I  DSN8   SYSDEFLT   CHANGED
****************************** END OF DB2 DATA******************************


Command ===>                                                     Scroll ===> CSR
```

Figure 6-49 Redefine Table Space (ADB21SAR)

Increasing the value for Numparts field from 0 to 2 adds two new lines (partitions) to the list, in which you can then modify or adjust the values for the new partitions. See Figure 6-50.

```
 DB2 ADMIN ------------------ DSN8 Redefine Table Space ------- Row 1 to 2 of 2

 Commands: CONTINUE  ORIGINAL  BALANCE  VALUES
 Line commands: S - Split part   R - Remove part  O - Original data
                C - Clear data
 CREATE TABLESPACE: TSREDBK  IN  DBREDBK    (Convert to Partitioned)

 Numparts       ===> 2           Large      ===> NO    LOB      ===>
 Define         ===> YES         DSSIZE     ===>       LOG      ===>
 Member Cluster ===> NO          SEGSIZE    ===>       CCSID    ===> EBCDIC
 Buffer Pool    ===> BP0         Close Rule ===> YES   Max Rows ===> 255
 Lock Size      ===> ANY         Lock Part  ===> NO    Lock Max ===> SYSTEM

                                Free  Pct       E T S
 S   Part       Pqty       Sqty Page Free Compr R M T VCAT     Stogroup
GBPCache
 - ------ ---------- ---------- ---- ---- ----- - - - ------- -------> --------
 Default:         48         48    0    0 YES   N Y I DSN8     SYSDEFLT CHANGED
       1          48         48    0    0 YES   N Y I DSN8     SYSDEFLT CHANGED
       2          48         48    0    0 YES   N Y I DSN8     SYSDEFLT CHANGED
 ****************************** END OF DB2 DATA******************************
```

Figure 6-50 Redefine Table Space (ADB21SAR)

If you enter the ORIGINAL command or place the cursor below the word, the table space is restored to its original values or resets all of your changes.

If you enter the CONTINUE command or place the cursor below the word, you are prompted to add values for the partition within the Create Partition panel (ADB21SAX). See Figure 6-51.

```
DB2 ADMIN -------------------- DSN8 Create Partitioni     Enter required field

Commands: CONTINUE  ORIGINAL  BALANCE  VALUES

CREATE INDEX  LUCYK    >  .                    >
          ON  LUCYK       . T1REDBK

Unique          ===>          Where Not Null  ===>      Cluster      ===>
Buffer pool     ===>          Close rule      ===>      Copy Allowed ===>
Piecesize       ===>          Define          ===>      Defer        ===>
                              Padded          ===>
( Column List ) ===>

         Primary    Secondary  Free Pct        S
S   Part Quantity   Quantity   Page Free Erase T VCAT     Stogroup GBPCache
- ------ ---------- ---------- ---- ---- ----- - -------- -------> --------
Default          12         12    0   10
       1         12         12    0   10

Command ===> VALUES                                          Scroll ===> CSR
```

Figure 6-51 Create Partition (ADB21SAX)

If you had entered BALANCE or placed the cursor below the word from the Redefine Table Space panel (ADB21SAR), the partitions would have been set to the average size. The total space is divided by the number of partitions and that value is used to allocate all partitions. See Figure 6-52.

```
DB2 ADMIN ----------------- DSN8 Redefine Table Space ------- Row 1 to 2 of 2

Commands: CONTINUE  ORIGINAL  BALANCE  VALUES
Line commands: S - Split part   R - Remove part  O - Original data
               C - Clear data
CREATE TABLESPACE: TSREDBK   IN  DBREDBK    (Convert to Partitioned)

Numparts       ===> 2          Large      ===> NO    LOB      ===>
Define         ===> YES        DSSIZE     ===>       LOG      ===>
Member Cluster ===> NO         SEGSIZE    ===>       CCSID    ===> EBCDIC
Buffer Pool    ===> BPO        Close Rule ===> YES   Max Rows ===> 255
Lock Size      ===> ANY        Lock Part  ===> NO    Lock Max ===> SYSTEM

                               Free  Pct       E T S
S   Part       Pqty       Sqty Page Free Compr R M T VCAT     Stogroup GBPCache
- ------ ---------- ---------- ---- ---- ----- - - - -------- -------> --------
Default:         48         48    0    0 YES   N Y I DSN8     SYSDEFLT CHANGED
      1                         0    0 YES   N Y I DSN8     SYSDEFLT CHANGED
      2                         0    0 YES   N Y I DSN8     SYSDEFLT CHANGED
****************************** END OF DB2 DATA ******************************

Command ===>                                                 Scroll ===> CSR
```

Figure 6-52 Redefine Table Space (ADB21SAR)

Archived

6.2 Modifying the Master Development Database (D0) using generated DDL and Object Compare

Some very important functions in a computer facility are creating new objects or making enhancements to existing objects. These changes normally move from the development system to the test system and are eventually implemented in multiple production sites. After the changes are made to the objects, you can now use DB2 Object Compare and CM to manage, track, synchronize, and carry out the changes.

Object Compare can use any of these sets of items:

- Two base version files
- Two DDL files
- Two catalog objects
- One DDL file and one catalog object

The use of ignore files and mask files can be used during the Object Compare process, which creates the Compare. The source and target files can be objects within the same system, like a development system. Another way to use Object Compare is to compare source and target files between two systems with source being the development objects and target being production system objects.

When using two different system objects, you would be doing a PROMOTE process and using OC to create a delta data set, which is made available from the Dev system (source) to the target system (that is, Test, Production) for the import data. Once the import is completed on the target system, the delta file (consisting of the changes) is registered in CM on the target system and can be analyzed and run to promote the changes to the target system objects.

We highly recommend following careful planning and change management strategies. Some database changes can have cascading effects on application performance and can be difficult to find and correct. The new CM process provides an interface to manage all of the version files for a database and allows you to track the history of changes. The use of change owner for every piece of work can provide a method for maintaining a relationship between version files, mask files, ignore files, generated DDL, and unloaded data. The DBA can query, check, track, and find all of the components involved in the changes.

6.2.1 Making changes to the database

Changes can be both logical and physical to the database implementation. You can create your own DDL or use DB2 Admin to create the DDL.

A few examples of *logical* changes to a database would be a change to the column name, length, or type.

A few examples of *physical* changes would be a change of a segmented table space to a partitioned table space, or a column added to a table, or non-compressed table space to a compressed table space, index changed from CLOSE YES to CLOSE NO, and adding a TRIGGER.

6.2.2 Generating and modifying DDL

Next we go through the steps to update changes to the database and use Object Compare with Change Management.

In this example, we want to make the physical changes shown in Table 6-1.

Table 6-1 Physical Changes

Object Name	From	To
Database DSG24D04	BUFFERPOOL BP15	BUFFERPOOL BP1
TABLESPACE GLWS001	COMPRESS NO	COMPRESS YES
UNIQUE INDEX DSG24D04.GLWXACT1	PRIQTY 12 SECQTY 12	PRIQTY 48 SECQTY 48
UNIQUE INDEX DSG24D04.GLWXACT1	FREEPAGE 0 PCTFREE 5	FREEPAGE 5 PCTFREE 10

We want to make the logical changes shown in Table 6-2.

Table 6-2 Logical Changes

Object Name	Column From	Column To
TABLE DSG24D04.GLWTACT	ACTKWD	ACTIONT
COLUMN ACTKWD / ACTIONT	CHAR(6) FOR SBCS DATA NOT NULL	CHAR(10) FOR SBCS DATA NOT NULL

These changes are in DDL with the name of PAOLOR4.SOURCE.DDL. In our example the objects are already part of the system (development). The target file is specified from the catalog from the development system.

Archived

6.2.3 Using DB2 Object Compare panels

You select Object Compare from the DB2 Administration Menu by entering **c** on the command line. See Figure 6-53.

```
DB2 Admin -------------- DB2 Administration Menu 7.2.0 ------------------ 15:46
Option ===> c

   1 - DB2 system catalog                                DB2 System: DB8A
   2 - Execute SQL statements                            DB2 SQL ID: PAOLOR4
   3 - DB2 performance queries                           Userid    : PAOLOR4
   4 - Change current SQL ID                             DB2 Rel   : 815
   5 - Utility generation using LISTDEFs and TEMPLATEs
   P - Change DB2 Admin parameters
  DD - Distributed DB2 systems
   E - Explain
   Z - DB2 system administration
  SM - Space management functions
   W - Manage work statement lists
   X - Exit DB2 Admin
  CC - DB2 catalog copy version maintenance
  CM - Change management

Interface to other DB2 products and offerings:
   C  DB2 Object Comparison Tool
   I  DB2I
  WL  Workload Generator
```

Figure 6-53 Administration Menu 7.2.0 (ADB2)

You are prompted to enter the source file attributes in the panel, DB2 Object Comparison Tool Menu. Since we have DDL, we enter **1** on the command line. See Figure 6-54.

```
COMPARE ---------------- DB2 Object Comparison Tool Menu ---------------- 15:49
Option ===> 1

                                            Status:
   1 - Specify compare source (new)         DDL file specified
   2 - Specify compare target (old)         DB2 catalog extract specified
   3 - Specify compare masks                None specified
   4 - Specify fields to ignore             Using defaults
   5 - Generate compare job                 Generated

   W - Walk through steps 1 - 5 in sequence
   V - Generate job to extract version file from source only
   R - Reset status

   S  - Save dialog
   M  - Manage/Restore dialog
   MC - MultiCompare
```

Figure 6-54 DB2 Object Comparison Tool Menu (GOCMENU)

Next you select option **1** from the panel, Specify Compare Source, to choose Source is from a DDL file. See Figure 6-55.

```
Compare -------------------- Specify Compare Source --------------------- 12:53
Option ===> 1

   1 - Source is from a DDL file
   2 - Source is from the DB2 catalog
   3 - Source is from a compare version file
```

Figure 6-55 Specify Compare Source (GOC1)

Next you add the DDL file name and data set name of the version file and any comments you want to add.

Important: The DDL file MUST include all dependent objects and authorizations. Otherwise, any subsequent comparison with the version file created by this Compare could lead to loss of these dependent objects.

In our case the DDL includes all dependent objects and authorizations. We supply the DDL file name of 'PAOLOR4.SOURCE.DDL', the version data set name of PAOLOR4.SOURCE.DDL.version1, and comments of DDL source DSG24D04 April 8. See Figure 6-56.

```
COMPARE ------------------- Specify Source DDL File --------------------------
Option ===>

Specify input DDL file:
  DDL data set . . 'PAOLOR4.SOURCE.DDL'

Specify compare version file output:
  Version table entry:
    Owner . . .          >           (? to look up)
    Name  . . .                  >   (? to look up)
  Data set:
    Data set name . PAOLOR4.SOURCE.DDL.vers1

Enter a description (optional):
  Description  . .  DDL source DSG24D04 April 8

-------------------------------------------------------------------------------
| ---------------------- DB2 Object Compare Warning ---------------------- |
| You have asked to generate a version file, but the source for these      |
| objects is a DDL file.  If this DDL does not include all dependent objects|
| and authorizations, any subsequent comparison with this version file     |
| may lead to loss of these dependent objects. Press  ENTER to continue    |
| or END to stop this operation.                                           |
|                          Please read this carefully                      |
-------------------------------------------------------------------------------
```

Figure 6-56 Specify Source DDL File (GOC11)

Once you press Enter, you are taken back to DB2 Object Comparison Tool Menu panel so you can enter **2** to specify the target file. Next you select **4** to have the objects automatically selected from the catalog. See Figure 6-57.

```
COMPARE -------------------- Specify Compare Target --------------------- 16:11
Option ===> 4


   1 - Target is from a DDL file
   2 - Target is from the DB2 catalog
   3 - Target is from a compare version file
   4 - Target is from the DB2 catalog and the objects are automatically
       selected based on the selected source objects
```

Figure 6-57 Specify Compare Target (GOC1)

Again you press Enter to be taken to the panel to specify the target information.

Important: Since the target objects are automatically selected based on the objects selected for the source, it is *very* important that you have all the dependent objects. Otherwise, if the source version file does not include all dependent objects, any subsequent comparison with the target might lead to a loss of these dependent objects. To prevent a loss of objects, select **Suppress DROP of objects** when you generate the Compare job(s).

You enter the location name (in this case **DB8A**), the compare version file to be a data set in **CM** of **PAOLOR4.TARGET.CAT.DSG24d04**, and a description, **cat target dsg24d04 April 8**. See Figure 6-58.

```
COMPARE ----------------- Specify Target DB2 Location -------------------------
Option ===>


Specify location name : DB8A            >

Specify output compare version file:
  Version data set : PAOLOR4.TARGT.CAT.DSG24d04

Enter a description (optional):
  Description  . . :  cat target dsg24d04 April 8

-------------------------------------------------------------------------------
| ----------------------- DB2 Object Compare Warning ---------------------- |
| Target objects will be automatically selected based on the objects you    |
| selected for the source. If the source version file does not include all  |
| dependent objects, any subsequent comparison with the target may lead to  |
| a loss of these dependent objects. To prevent a loss of objects, select   |
| "Suppress DROP of objects" when you generate the compare job(s).          |
| Press ENTER to continue or END to stop this operation.                    |
|                          Please read this carefully                       |
-------------------------------------------------------------------------------
```

Figure 6-58 Specify Target DB2 Location (GOC14)

Once you press Enter, the DB2 Object Compare Tool Menu panel comes back. You are now ready to generate the Compare job. You enter **5** on the command line. See Figure 6-59.

```
COMPARE ---------------- DB2 Object Comparison Tool Menu ---------------- 19:11
Option ===> 5

                                                Status:
   1 - Specify compare source (new)             DDL file specified
   2 - Specify compare target (old)             Automatic (DB2 catalog extract)
   3 - Specify compare masks                    None specified
   4 - Specify fields to ignore                 Using defaults
   5 - Generate compare job                     Generated

   W - Walk through steps 1 - 5 in sequence
   V - Generate job to extract version file from source only
   R - Reset status

   S  - Save dialog
   M  - Manage/Restore dialog
   MC - MultiCompare
```

Figure 6-59 DB2 Object Comparison Tool Menu (GOCMENU)

You should fill in the Worklist name, data set information for PDS for batch jobs, prefix for data set, and any other option for the GOC run. See *IBM DB2 Object Comparison Tool for z/OS User's Guide Version 7 Release 2,* SC18-9554 for details regarding the available options. In this case we are cautious and specify Suppress DROP of objects = **yes**. See Figure 6-60.

Archived

```
COMPARE -------------------- Generate Compare Jobs  ---------------------------
Option ===>

 Specify the following for DB2 Object Comparison Tool:
                                                                      More:     +
 Worklist information:
   Worklist name . . . . . . . paolor4b  (also used as middle qualifier in
DSNs)

 Compare options:
   Suppress DROP of objects  . yes      (Yes/No)
   Suppress DROP of columns  . NO       (Yes/No)
   Suppress adding columns . . NO       (Yes/No)
   Run SQLID . . . . . . . . .          (Blank, an SQLID, or <NONE>)

 Change reporting options  . . NO       (Yes/No)

 Data set information:
   PDS for batch jobs  . . . . PAOLOR4b.PDS
   Prefix for data sets  . . . PAOLOR4b

 Options:
   Single compare job  . . . . YES      (Yes/No)
     Member name . . . . . . . COMPARE  (default COMPARE )
   Generate apply jobs . . . . YES      (Yes/No)
     Generate one job  . . . .          (Yes, No, or (Per) Process)
       Member prefix . . . . . APPLY    (default APPLY )
     As work statement list  . YES      (Yes/No to append to work stmt list)
     Use utility options . . . no       (Yes/No)
     Unload Method . . . . . . U        (Unload, Parallel unload)
     Generate templates. . . . NO       (Yes/No)
     Stop on conversion error. no       (Yes/No)

     Optional jobs after Reload or Alter:
       Run CHECK DATA  . . . . NO       (Yes/No)
       Take an image copy  . . N        (after: Reload/Alter/Both/None)
       Run REORG/REBUILD . . . N        (Mandatory, All relevant, None)
       Run RUNSTATS  . . . . . N        (after: Reload/Alter/Both/None)
       Run REBIND  . . . . . . NO       (Yes/No)

 BP - Change batch job parameters
 TU - Specify TEMPLATE usage
 UO - Change utility options
```

Figure 6-60 Generate Compare Jobs (GOC5)

Important: It is a safer option to specify **Yes** on Suppress DROP of objects in case something was left out of the source. This prevents dropping objects not found in the source but present in the target catalog.

Archived

6.2.4 Running a Compare job and using the CM function

You press Enter from the Generate Compare Jobs panel after you have entered all of the desired options. You are then prompted to register this change in the Change Management Prompt panel. You should enter **Yes**. See Figure 6-61.

```
DB2 ADMIN ----------- DB8A Change Management Prompt ----------- 20:52

Change Management is optional for SQLID:  PAOLOR4

Do you wish to use Change Management for this function:  YES(Yes/No)
```

Figure 6-61 Change Management Prompt (ADB2CMRO)

6.2.5 Registering a CHANGE

From the CM prompt panel you press Enter and are allowed to register the change name and comments. Notice that this is a Change Type of Change or a normal change. See Figure 6-62.

```
DB2 Admin -------------------- CM - Register Options ------------------- 20:56
Option ===>

Commands: CONTINUE                                DB2 System: DB8A
                                                  DB2 SQL ID: PAOLOR4
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR4  >  (Optional, default is PAOLOR4)
Name  . . . . . . . . GOC2dsg24d04                                        >
Comment . . . . . . . goc april 8 changes to dsg24d04                     >
Change Type . . . . : CHANGE   (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                      Owner         Name
Ignore  . . . . . . .         >                           >  (? to look up)
Mask  . . . . . . . .         >                           >  (? to look up)
Delta Version . . . .         >                           >  (? to look up)
```

Figure 6-62 CM - Register Options (ADB2CRO)

Once you press Enter, you are prompted to decide what to do if there are pending changes. In this example we specify supersede. See Figure 6-63.

```
COMPARE ---------------- Specify Register Mode ---------------- 21:01

Action if pending changes  supersede (Cancel/Prereq/Supersede)
```

Figure 6-63 Specify Register Mode (GOC5RM)

Once you press Enter, you see the job displayed and can enter **sub** on the command line to submit the job. The job output shows the various changes and also that the other objects had no changes. See Figure 6-64.

```
OBJECT COMPARISON REPORT
========================
          Source:
                  Extracted from location *FROM DDL FILE*  at 2007-03-23 12:45
                       by PAOLOR4

          Target: CAT TARGET DSG24DO4 APRIL 8
                  Extracted from DB8A at 2007-03-23 12:45 by PAOLOR4

          Target system is DB2 Release 810

Compare database source(DSG24DO4) and target(DSG24DO4)
    (A)Field BUFFERPOOL changed from BP15 to BP1
  Database will be altered

Compare tablespace source(DSG24DO4.GLWSDPT) and target(DSG24DO4.GLWSDPT)
  No changes to Tablespace
.
.
.
Compare tablespace source(DSG24DO4.GLWS001) and target(DSG24DO4.GLWS001)
    (A)Field COMPRESS changed from NO to YES
  Tablespace will be altered

Compare tablespace source(DSG24DO4.GLWS002) and target(DSG24DO4.GLWS002)
  No changes to Tablespace

Compare table source(DSG24DO4.GLWTACT) and target(DSG24DO4.GLWTACT)
  (A)Column ACTIONT inserted
  (A)Column ACTKWD dropped
  Table DSG24DO4.GLWTACT will be dropped
  Table will be recreated
  Table data will be converted

Compare table source(DSG24DO4.GLWTDNG) and target(DSG24DO4.GLWTDNG)
  No changes to Table
.
.
.
Compare index source(DSG24DO4.GLWXACT1) and target(DSG24DO4.GLWXACT1)
    (A)Field PRIQTY changed from 12 to 48
    (A)Field SECQTY changed from 12 to 48
    (A)Field FREEPAGE changed from 0 to 5
    (A)Field PCTFREE changed from 5 to 10
  Index will be altered

Compare index source(DSG24DO4.GLWXDNG1) and target(DSG24DO4.GLWXDNG1)
  No changes to Index
.
.
.
GOC2CMP - Ended normally
```

Figure 6-64 job output

We also see this new change registered in CM. See Figure 6-65.

```
 DB2 Admin ------------------ CM - Changes -------------------- Row 1 to 8 of 8
 Command ===>                                                   Scroll ===> PAGE

 Line commands:
  U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
  PQ - Prerequisites  IG - Ignores  MA - Masks
  ? - Show all line commands

 Sel       ID Owner    Name               Type    Status   Comment
            * *        *                  *       *        *
 --- -------- -------- ------------------ ------- -------- ------------------->
          233 PAOLOR4  ADCOL1GWLTEMP      CHANGE  COMPLETE AL ADDCOL1 CHAR 12 G
          290 PAOLOR4  ALCPRIMKEY1        CHANGE  COMPLETE ALC PRIMKEY DEPT_NO
          292 PAOLOR4  RALCPRIMKEY        RECOVER ANALYZED RECOVER CHANGE FOR H
          293 PAOLOR4  GOCDSG24D04        CHANGE  DEFINED  GOC APRIL 8 CHANGES
****************************** END OF DB2 DATAA ******************************
```

Figure 6-65 Changes (ADB2C11)

6.2.6 ANALYZE and RUN

Once the Object Compare job completes, the change can be analyzed and run as a normal change within CM.

Another way to use Object Compare is to promote changes. You can create a delta file of the changes that Object Compare found when the GOC job runs. The delta file is stored in the source system's CM database and can then be imported and registered as a change on the target system, which can then be analyzed and run within CM.

6.2.7 Specify Ignore Fields (optional)

The use of an ignore file within Object Compare allows you to specify that certain fields in the DB2 catalog records are to be ignored when objects are compared. An ignore file is used when GOC is analyzing the changes. You want to compare the source changes to the target objects but also want to maintain certain values in the target object. An example might be that you want to preserve certain physical fields like primary and secondary space values or buffer pools for the database. Ignoring these physical fields by selecting these options in the ignore file tells Object Compare to ignore and not compare these fields in both the source and target files. Therefore DB2 Object Compare does not try to change these values and thereby preserves them.

> **Note:** Ignore fields are applied to both the target and the source objects before the definitions are compared. No field in the DB2 catalog record for which an ignore is specified is compared. If an object is dropped and re-created because of other changes, values for ignored fields are taken from the target version. All other fields have values from the source version.

Some catalog fields are automatically ignored, such as statistics, dates, and internal identifiers. These are called system ignores and are automatically included by default.

Those fields that you can select are used to protect or preserve the fields from updates. You might want to use this when comparing test objects to the production system objects.

Table 6-3 shows the available Ignore fields.

Table 6-3 The DB2 catalog table ignore fields

DB2 catalog table	Ignore fields
SYSCHECKS	CREATOR, CHECKCONDITION
SYSCOLUMNS	COLTYPE, LENGTH, SCALE, NULLS, REMARKS, DEFAULT, KEYSEQ, FOREIGNKEY, FLDPROC, LABEL, DEFAULTVALUE, LENGTH2, TYPESCHEMA, TYPENAME, STATS_FORMAT, PARTKEY_COLSEQ, PARTKEY_ORDERING, ALTEREDTS
SYSDATABASE	CREATOR, STGROUP, BPOOL, ROSHARE, TYPE, GROUP_MEMBER, ENCODING_SCHEME, SBCS_CCSID, DBCS_CCSID, MIXED_CCSID, INDEXBP
SYSDATATYPES	OWNER, SOURCESCHEMA, SOURCETYPE, METATYPE, LENGTH, SCALE, SUBTYPE, ENCODING_SCHEME, REMARKS
SYSFIELDS	FLDPROC, WORKAREA, EXITPARML, PARMLIST, EXITPARM
SYSINDEXES	UNIQUERULE, CLUSTERING, BPOOL, PGSIZE, ERASERULE, DSETPASS, CLOSERULE, INDEXTYPE, PIECESIZE, COPY, SPACEF, REMARKS, PADDED, VERSION, OLDEST_VERSION, CURRENT_VERSION, RELCREATED, AVGKEYLEN
SYSINDEXPART	PARTITION, PQTY, SQTY, STORTYPE, STORNAME, VCATNAME, LIMITKEY, FREEPAGE, PCTFREE, INDEXTYPE, GBPCACHE, SECQTYI, SPACEF, DSNUM, EXTENTS, PSEUDO_DEL_ENTRIES, LEAFNEAR, LEAFFAR
SYSSYSKEYS	COLSEQ, ORDERING
SYSPARMS	OWNER, SPECIFICNAME, CAST_FUNCTION, PARMNAME, ROWTYPE, ORDINAL, TYPESCHEMA, TYPENAME, LOCATOR, TABLE, TABLE_COLNO, LENGTH, SCALE, SUBTYPE, CCSID, ENCODING_SCHEME
SYSRELS	RELNAME, DELETERULE, IXOWNER, IXNAME, ENFORCED, CHECKEXISTINGDATA
SYSROUTINES	OWNER, CREATEDBY, SPECIFICNAME, RETURN_TYPE, ORIGIN, FUNCTION_TYPE, PARM_COUNT, LANGUAGE, COLLID, SOURCESCHEMA, SOURCESPECIFIC, DETERMINISTIC, EXTERNAL_ACTION, NULL_CALL, CAST_FUNCTION, SCRATCHPAD, SCRATCHPAD_LENGTH, FINAL_CALL
SYSSEQUENCES	OWNER, SEQTYPE, INCREMENT, START, MAXVALUE, MINVALUE, CYCLE, CACHE, ORDER, REMARKS, PRECISION, RESTARTWITH
SYSTABLEPART	IXNAME, IXCREATOR, PQTY, SQTY, STORTYPE, STORNAME, VCATNAME, LIMITKEY, FREEPAGE, PCTFREE, COMPRESS, GBPCACHE, TRACKMOD, SECQTYI, SPACEF, DSNUM, EXTENTS, LOGICAL_PART
SYSTABLES	TYPE, DBNAME, TSNAME, EDPROC, VALPROC, CLUSTERTYPE, REMARKS, KEYCOLUMNS, STATUS, LABEL, AUDITING, CREATEDBY, LOCATION, TBCREATOR, TBNAME, DATACAPTURE, CHECKS, ENCODING_SCHEME

DB2 catalog table	Ignore fields
SYSTABLESPACES	CREATOR, BPOOL, PARTITIONS, LOCKRULE, PGSIZE, ERASERULE, STATUS, IMPLICIT, DSETPASS, CLOSERULE, SEGSIZE, LOCKMAX, TYPE, ENCODING_SCHEME, SBCS_CCSID, DBCS_CCSID, MIXED_CCSID, MAXROWS
SYSTRIGGER	OWNER, TRIGTIME, TRIGEVENT, GRANULARITY, TEXT, REMARKS, TRIGNAME
SYSVIEWS	CHECK, TEXT, PATHSCHEMAS, RELCREATED, TYPE, REFRESH, ENABLE, MAINTENANCE, REFRESH_TIME, ISOLATION, SIGNATURE, APP_ENCODING_CCSID

Ignores that are specified when analyzing a change must be defined in the Change Management database, where the ignore is stored in a table.

Ignores that you specific when you explicitly use the DB2 Object Compare Tool to generate a Compare job can either be in the Change Management database or in a data set.

To create an ignore file, select option **3** on the Change Management (CM) panel to display the Manage Ignores panel. See Figure 6-66.

```
DB2 ADMIN -------------------- Change Management (CM) ------------------ 16:28
Option ===> 3

   1 - Manage changes                                DB2 System: DSN8
   2 - Manage masks                                  DB2 SQL ID: LUCYK
   3 - Manage ignores
   4 - Manage versions
   5 - Manage ID table
   6 - Report changes
```

Figure 6-66 Change Management (CM) (ADB2C)

Select option **2** on the Manage Ignores panel to create an ignore. See Figure 6-67.

```
DB2 ADMIN ------------------- CM - Manage Ignores ----------------------- 16:30
Option ===> 2

                                                                 More:     +
   1 - Display ignores                               DB2 System: DSN8
   2 - Create an ignore                              DB2 SQL ID: LUCYK
```

Figure 6-67 CM - Manage Ignores (ADB2C3)

Archived

Specify an owner and a name for the ignore, and optionally enter a comment for the ignore. See Figure 6-68.

```
DB2 ADMIN ---------------- CM - Insert Ignore --------------------------- 16:33
Command ===>


Owner . . . PAOLOR4  >  (Optional, default is LUCYK)
Name  . . . REDBKIGNORE        >  (Required, ? to look up)
Comment . . redbk ignore                                >
```

Figure 6-68 CM - Insert Ignore (ADB2C22)

Press PF3 to return to the Manage Ignores panel. Select option **1** to display the ignores on the Ignores Panel. See Figure 6-69.

Issue the **IL** line command for the ignore that you just created to define the ignore fields.

```
DB2 ADMIN ------------------ CM - Ignores -------------------- Row 1 to 6 of 6

Line commands: IL
 U - Update  DEL - Delete  INS - Insert  IL - Ignore lines  CH - Changes
 I - Details on ignore

Sel          ID Owner    Name              Comment
              * *        *                 *
--- ----------- -------- ----------------- ---------------------------------
             64 PAOLOR4  REDBKIGNORE       REDBK IGNORE
****************************** END OF DB2 DATA ******************************
```

Figure 6-69 CM - Ignores (ADB2C31)

Archived

This takes you to Specify Ignore Fields: Objects panel, as shown in Figure 6-70.

```
COMPARE ---------- Specify Ignore Fields: Objects ----------- Row 1 to 12 of 18

Valid line commands are:
 U - Update Ignore Fields

Select Object          Ignore Fields
       *               *
------ --------------- -------------------------------------------------------
       GENERIC         None
       SYSCHECKS       None
       SYSCOLUMNS      None
       SYSDATABASE     None
       SYSDATATYPES    None
       SYSFIELDS       None
       SYSINDEXES      None
       SYSINDEXPART    None
       SYSKEYS         None
       SYSPARMS        None
       SYSRELS         None
       SYSROUTINES     None
       SYSSEQUENCES    None
       SYSTABLEPART    None
       SYSTABLES       None
       SYSTABLESPACE   None
       SYSTRIGGERS     None
       SYSVIEWS        None
****************************** END OF DB2 DATA ******************************
```

Figure 6-70 Specify Ignore Fields: Objects (GOCCI)

You can select which object to ignore by placing a **U** by the object name and then you are prompted with a list for the individual Ignore Fields for the object. You then use **U** and **S** line commands to select or de-select a particular field to be ignored. Once you have selected all you want, you press PF3 to go back to the main menu.

You can delete an ignore file in CM by using the **DEL** line command while in the CM - Ignores panel.

GENERIC ignore field specifications provide a shortcut for ignoring all buffer pools, allocated space information, and information about how data is stored and partitioned. The generic ignore specifications are:

- BUFFERPOOL
- SPACE
- STORAGE
- PARTITIONING

For further information, see the section “Ignores” in Chapter 20 of the *IBM DB2 Administration Tools for z/OS User’s Guide and Reference,* SC18-9552.

Archived

6.2.8 Specify Mask Fields (optional)

DB2 Compare also can use masks. Masking enables you to change the names of the DB2 objects, owners, and schemas. Masking is often useful when the new WSL created by Object Compare is to be used on a different DB2 subsystem or in a different database. Masks specify how names are to be translated when objects are compared or when they are moved from one system to another (source to target). Change Management can now manage and define the masks.

A mask (also called translation mask) provides the ability to cause context-sensitive global changes to naming conventions when you use DB2 Object Comparison Tool and Admin Tool. Masks that are specified when you import changes through Change Management must be defined in the Change Management database, where the masks are stored in a table. Masks that are used for reverse engineering the catalog, cloning WSLs, migrating objects, or explicitly performing comparisons by using DB2 Object Comparison Tool can be defined in the Change Management database or in a data set. We recommend you to place all your masks in the Change Management database.

The masking capability of DB2 Object Compare is a very powerful feature. It allows you to specify universal change to identifiers like OWNER, SCHEMA, or NAME for all source objects before comparing to the target. Wild cards are also allowed, making the masks more general purpose.

Tip: Masks are applied to the source object *before* they are compared with the target. You can have as many masks as you want. However, having many masks can impact the performance of the Compare process.

The first left-hand mask that matches is used, and the name is translated to the right-hand value. If no match is found, the name is not translated but it still participates in the Compare.

For more information about how masks, see *IBM DB2 Administration Tool for z/OS User's Guide and Reference Version 7 Release 2,* SC18-9552.

To use CM to manage Masks, select option **2** on Change Management panel. This allows you to display the masks that are defined or the system and you create a new mask.

See Figure 6-71

```
DB2 ADMIN -------------------- Change Management (CM) ------------------- 16:28
Option ===> 2

   1 - Manage changes                                    DB2 System: DSN8
   2 - Manage masks                                      DB2 SQL ID: LUCYK
   3 - Manage ignores
   4 - Manage versions
   5 - Manage ID table
   6 - Report changes
```

Figure 6-71 Change Management (CM) (ADB2c)

To create a new mask, you choose option **2** on the CM - Manage Masks panel. See Figure 6-72.

```
DB2 Admin ------------------- CM - Manage Masks   ----------------------- 19:15
Option ===> 2


   1 - Display masks                                      DB2 System: DB8A
   2 - Create a mask                                      DB2 SQL ID: PAOLOR4









Enter display selection criteria (Using a LIKE operator, criteria not saved):
Name . . . . . . .                    >  Created by . .          >
Owner  . . . . . .          >            Altered by . .          >
Created within . .                       Mask ID  . . .
Altered within . .
```

Figure 6-72 CM - Manage Masks (ADB2C2)

Next you specify the owner of the mask, its name, and any comments for the new mask. See Figure 6-73.

```
DB2 ADMIN ---------------- CM - Insert Mask ----------------------------- 19:18
Command ===>


Owner . . . PAOLOR4  >  (Optional, default is PAOLOR4)
Name  . . . Redbkmask1         >  (Required, ? to look up)
Comment . . example of mask1                            >
```

Figure 6-73 CM - Insert Mask (ADB2C22)

Once you press Enter, you see the message, `INSERT stmt executed` on the CM - Insert Mask panel. See Figure 6-74.

```
ADB2C22 n ---------------- CM - Insert Mask ----------------------------- 19:21
Command ===>
INSERT stmt executed

Owner . . . PAOLOR4  >  (Optional, default is PAOLOR4)
Name  . . . REDBKMASK1         >  (Required, ? to look up)
Comment . . EXAMPLE OF MASK1                            >
```

Figure 6-74 CM - Insert Mask (ADB2C22)

You can now press PF3 and get back to the CM - Manage Mask panel. You can then choose option **1** to display the new mask data set. See Figure 6-75.

```
DB2 ADMIN ------------------- CM - Masks ------------------- Row 1 to 15 of 15
Command ===>                                                  Scroll ===> PAGE

Line commands:
 U - Update  DEL - Delete  INS - Insert  ML - Mask lines  CH - Changes
 E - Edit  I - Details on mask

Sel          ID Owner    Name                Comment
              * *        *                   *
--- ----------- -------- ------------------ --------------------------------->
             68 PAOLOR2  RENAME
             69 PAOLOR1  PROD
             81 PAOLOR4  REDBKMASK1          EXAMPLE OF MASK1
****************************** END OF DB2 DATA ******************************
```

Figure 6-75 CM - Masks (ADB2C21)

Important: Masks that have been created in an explicitly named data set outside of Change Management are not displayed because they are not stored in the Change Management database. You might have created a mask that was used when performing comparisons using DB2 Object Comparison Tool or other functions in DB2 Admin (such as reverse engineering, migrating DB2 data, or cloning work statement lists) in a data set. When you are prompted to specify the mask to use, you have the option of using masks that are either in data sets or in the Change Management database

Now you can enter the **ML** line command for the new mask just created to add the mask line definitions for the mask. For each mask line that you add, specify:

- The type of object for the mask in the Type field. For example, TBNAME specifies a mask for tables.
- The input mask (the pattern of the string that you want to translate) in the From field.
- The output mask (the string to which you want to translate) in the To field.

For example, to define a mask that translates any table name that starts with REDBK to a name that starts with TEST and a column name from DEPT to DPTN, enter the values that are shown in Figure 6-76.

```
DB2 ADMIN ---------------- CM - Mask Lines --------------------- Row 1 from 2
Command ===> SAVE                                              Scroll ===> PAGE

Mask lines for mask "PAOLOR4"."REDBKMASK1"
Commands: SAVE
Line commands:
  I - Insert  D - Delete  R - Repeat  M - Move  A - After  B - Before

Sel    Sequence Type     From                  To                     Oper.  T
              * *        *                     *                      *      *
--- ----------- -------- --------------------> ---------------------> ------ -

*             1 TBNAME   REDBK*                TEST*                  UPDATE
*             2 COLNAME  DEPT*                 DPTN*                  UPDATE
****************************** END OF DB2 DATA ******************************
```

Figure 6-76 Mask Lines (ADB2C2L)

You can use the **I**, **D**, and **R** line commands to quickly insert, delete, and repeat mask lines when you define a mask. You can also use the **A** and **B** line commands with the **M** line command to quickly move the mask lines around in the definition. The order of the mask lines in the definition is important because the first mask that matches is used and the name is translated to the second value. You should put the most specific translation masks at the beginning and the more general ones at the end. Once all the values have been entered, you enter **SAVE** or place the cursor under the word Save to complete the mask.

Now the new mask can be used in the DB2 Object Compare Tool or the DB2 Admin Tool.

When you register a change, you can specify the name of the mask in the CM - Register Options panel. See Figure 6-77.

```
DB2 Admin -------------------- CM - Register Options -------------------- 20:56
Option ===>

Commands: CONTINUE                                      DB2 System: DB8A
                                                        DB2 SQL ID: PAOLOR4
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR4  >  (Optional, default is PAOLOR4)
Name  . . . . . . . . GOC2dsg24d04                                        >
Comment . . . . . . . goc april 8 changes to dsg24d04                     >
Change Type . . . . : CHANGE   (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                      Owner         Name
Ignore  . . . . . . .         >                          >  (? to look up)
Mask  . . . . . . . . PAOLOR4  > REDBKMASK1              >  (? to look up)
Delta Version . . . .         >                          >  (? to look up)
```

Figure 6-77 CM - Register Options (ADB2CRO)

For information on mask definitions, see *IBM DB2 Administration Tool for z/OS User's Guide and Reference Version 7 Release 2,* SC18-9552.

Archived

Archived

7

Rolling out new versions of a database design

In this chapter we describe the two different ways you can deploy your new database design to another DB2 environment. This chapter is based on the changes you implemented described in Chapter 6, “Implementing changes to a design (Vn)” on page 257.

You can do the rollout of a new database design by either using the delta change data set provided by Compare or by using the delta change data set provided by the PROMOTE process.

The IMPORT process after using the PROMOTE process or after using Compare is exactly the same. Therefore, this process is described only once in 7.3, “Importing delta changes data set” on page 327 and is not repeated for the PROMOTE process and the Compare process.

© Copyright IBM Corp. 2007. All rights reserved.

7.1 Rolling out using the PROMOTE process

When rolling out using the PROMOTE process, we assume the following conditions:

- A base version existed before the change was done (starting base version).
- The change was done.
- A base version existed after the change was done (ending base version).

The PROMOTE process identifies the differences between two base versions. The goal is to get a delta between two versions, one representing a starting version before a change has been made, one representing an ending version after a change has been made. The delta identified by the PROMOTE process is called a *delta changes data set*. See Figure 7-1.

For further information regarding versions in general and the PROMOTE process, refer also to 4.12, “Versions” on page 186 and 4.4.5, “The PROMOTE process” on page 151.

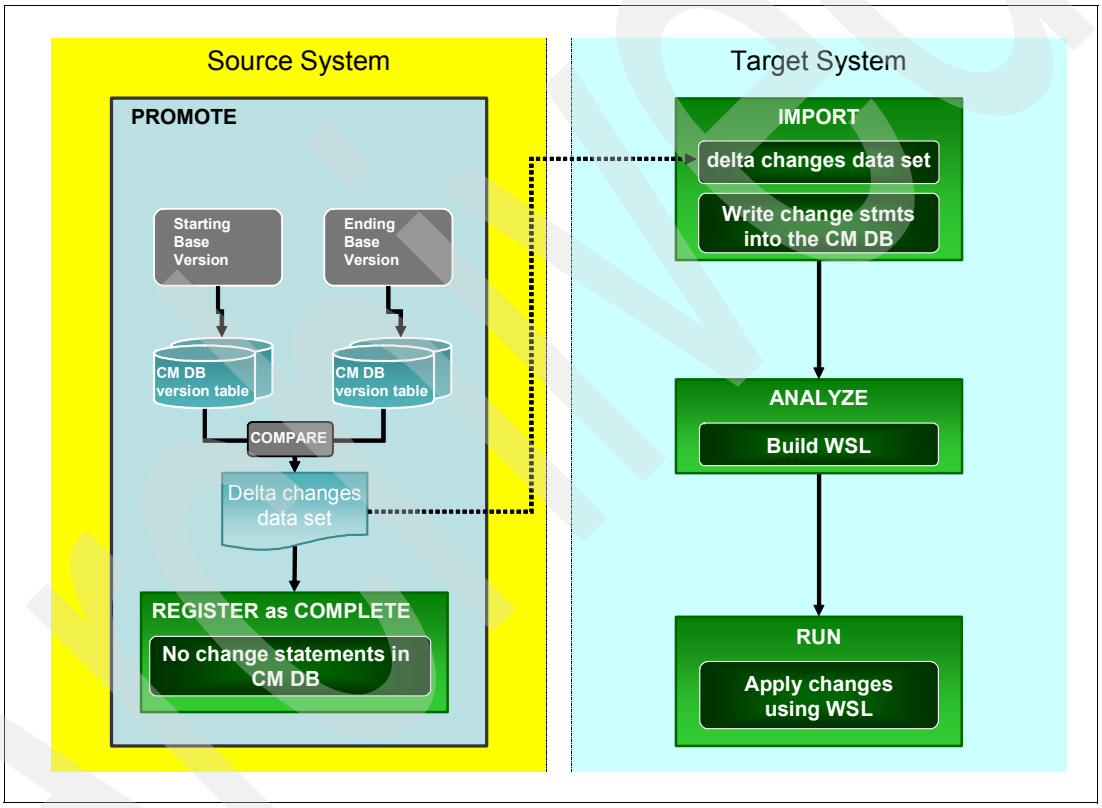

Figure 7-1 Rolling out using PROMOTE

7.1.1 The change being made

With this example we want to promote a changed column size for table DSG24D01.GLWTEPA. The column UPDATED_BY is supposed to be changed from CHAR(8) to CHAR(80).

7.1.2 Comparing the base versions

First check if you have a starting and an ending base version for your change. Go to **4** - Manage versions from the Change Management Menu as shown in Figure 7-2.

```
DB2 Admin -------------------- Change Management (CM) ------------------- 21:11
Option ===> 4


   1 - Manage changes                                    DB2 System: DB8A
   2 - Manage masks                                      DB2 SQL ID: PAOLOR1
   3 - Manage ignores
   4 - Manage versions
   5 - Manage ID table
   6 - Report changes
```

Figure 7-2 Manage versions

Then display your version scope as in Figure 7-3.

```
DB2 Admin ------------------- CM - Manage Versions ---------------------- 21:12
Option ===> 2


   1 - Display versions                                  DB2 System: DB8A
   2 - Display version scopes                            DB2 SQL ID: PAOLOR1
   3 - Insert a version scope









Enter display selection criteria (Using a LIKE operator, criteria saved):
Name . . . . . . . DSG%              >  Created by . .          >
Owner  . . . . . . PAOLOR1% >           Altered by . .          >
Created within . .                      Version ID . .
Altered within . .
```

Figure 7-3 Looking up your version scope

Archived

With the line command **VE** you can look up your existing versions. Notice that you must have base versions displayed. Only base versions are valid for the base version input for the PROMOTE process. See Figure 7-4.

```
DB2 Admin --------------- CM - Version Scopes --------------------- Row 1 of 1
Command ===>                                                  Scroll ===> CSR

Line commands:
 VE - Versions  SO - Version scope objects  GV - Generate new base version
 INS - Insert  U - Update  DEL - Delete  I - Details on version scope

Sel          ID Owner    Name                Comment
              * *        *                   *
--- ----------- -------- ------------------ ----------------------------------
VE           29 PAOLOR1  DSG24D01           DSG24D01
****************************** END OF DB2 DATA ******************************
```

Figure 7-4 Looking up the versions for a version scope

In our example, there are two versions displayed. V0 has been taken before a change occurred, V1 is the base version after a change has been done. See Figure 7-5.

```
DB2 Admin ------------------ CM - Versions ------------------------ Row 1 of 2
Command ===>                                                  Scroll ===> CSR

Line commands:
 CH - Changes  PR - Promote  VS - Version scope  DEL - Delete  U - Update
 PT - Toggle protected status  I - Details on version

Sel          ID T Owner    Name                Comment
              * * *        *                   *
--- ----------- - -------- ------------------ -------------------------------
            238 B PAOLOR1  DSG24D01 V1
            236 B PAOLOR1  DSG24D01 V0
****************************** END OF DB2 DATA ******************************
```

Figure 7-5 Base versions

Notice that the column T (type) has to be **B** for both versions. **B** is the marker for a base version.

We then do a line command in front of the ending base version, which is **PR** for the PROMOTE process. In our case, the ending base version is DSG24D01 V1. The version name from the line where the **PR** has been entered is automatically taken as an ending base version in the following panel. So you just have to enter the starting version. In our case, the starting version is **DSG24D01 V0**. See Figure 7-6.

Remember that every version is stored internally within the Change Management database. You are taken from :cmown.ADBCVERSION and :cmown.ADBCVERLINES.

```
DB2 Admin ----------------------- CM - Promote -------------------------- 21:37
Command ===>

Starting Version (Old):
  Owner . . . . . . PAOLOR1  >            (? to look up)
  Name  . . . . . . DSG24D01 V0        >  (? to look up)

 or enter a data set name that contains a Start Version:

  Data set name . .

Ending Version (New):
  Owner . . . . . . PAOLOR1  >            (? to look up)
  Name  . . . . . . DSG24D01 V1        >  (? to look up)

Output data set names:
  Promote JOB JCL . 'ADMIN.CM.CNTL.PROMOTE'
  Delta change  . . 'ADMIN.CM.DELTA'
```

Figure 7-6 The PROMOTE process

7.1.3 Registering the change

After providing the base version information and the data set to save the delta between the versions, you are asked to register a change. Enter a unique name for your change and point-and-shoot to CONTINUE. You do not have to specify a delta version. See Figure 7-7.

```
DB2 Admin -------------------- CM - Register Options -------------------- 21:48
Option ===>

Commands: CONTINUE                                     DB2 System: DB8A
                                                       DB2 SQL ID: PAOLOR1
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR1  >  (Optional, default is PAOLOR1)
Name  . . . . . . . . D070316-CR#92-NOR                                  >
Comment . . . . . . .                                                    >
Change Type . . . . : COMPARE  (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                      Owner         Name
Ignore  . . . . . . .         >                          >  (? to look up)
Mask  . . . . . . . .         >                          >  (? to look up)
Delta Version . . . .         >                          >  (? to look up)
```

Figure 7-7 Register a change for the PROMOTE process

Archived

There is a PROMOTE job generated in the data set **ADMIN.CNTL.CM.PROMOTE** that has to be submitted. If the PROMOTE job runs successfully, your recently registered change changes its status to COMPLETE. See Figure 7-8.

```
DB2 Admin ------------------ CM - Changes ------------------------- Row 1 of 1
Command ===>                                                   Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- --------------------
I        271 PAOLOR1  D070316-CR#92-NOR  COMPARE COMPLETE
****************************** END OF DB2 DATA ******************************
```

Figure 7-8 The status of a change after doing the PROMOTE process

The line command **I** can be entered and you are prompted with additional information such as the status and the delta changes data set. See Figure 7-9.

```
DB2 Admin ------------- CM - Interpretation of a Change ---------------- 12:55
Command ===>

Details for change : PAOLOR1.D070316-CR#92-NOR

Change ID  . . . : 271            Status . . . : COMPLETE
Created by . . . : PAOLOR1        Created  . . : 2007-03-20-21.48.37.367064
Last altered by  : PAOLOR1        Last altered : 2007-03-20-21.48.56.902434
Change type  . . : COMPARE
Delta change DSN : ADMIN.CM.DELTA
Promote job DSN  : ADMIN.CM.CNTL.PROMOTE

Comment  . . . . :



                   Owner         Name
Change . . . . . . : PAOLOR1     : D070316-CR#92-NOR
Mask . . . . . . . : ?           : ?
Ignore . . . . . . : ?           : ?
Version scope. . . : ?           : ?
Base version . . . : ?           : ?
Source version . . : PAOLOR1     : DSG24D01 V1
Target version . . : PAOLOR1     : DSG24D01 V0
Delta version. . . : ?           : ?
```

Figure 7-9 Information about the PROMOTE process

The PROMOTE job finally does the version compare between the starting and ending version and builds the data set **ADMIN.CM.DELTA**. This is the delta changes data set built from the PROMOTE process.

The delta changes data set contains the DDL as shown in Figure 7-10.

```
 BROWSE    ADMIN.CM.DELTA                              Line 00000000 Col 001 080
 Command ===>                                                   Scroll ===> CSR
******************************** Top of Data ********************************
CREATE    ALTER TABLE DSG24D01.GLWTEPA
CREATE      ALTER UPDATED_BY SET DATA TYPE CHAR(80) ;
CREATE    COMMIT ;
****************************** Bottom of Data ******************************
```

Figure 7-10 Content of the delta changes data set

You can use this data set for the IMPORT process in another DB2 environment. The IMPORT process is described in 7.3, “Importing delta changes data set” on page 327 and in terms of general information in 4.4.6, “The IMPORT process” on page 155.

7.2 Rolling out using Compare

When rolling out using DB2 Compare, we assume the following conditions:

- There is a source object available that contains the change.
- There is a target object available that does not contain the change.

We use DB2 Compare to identify differences between two tables given as a source and as a target. The goal is to get the differences between source and target and to write them into a delta changes data set. See Figure 7-11.

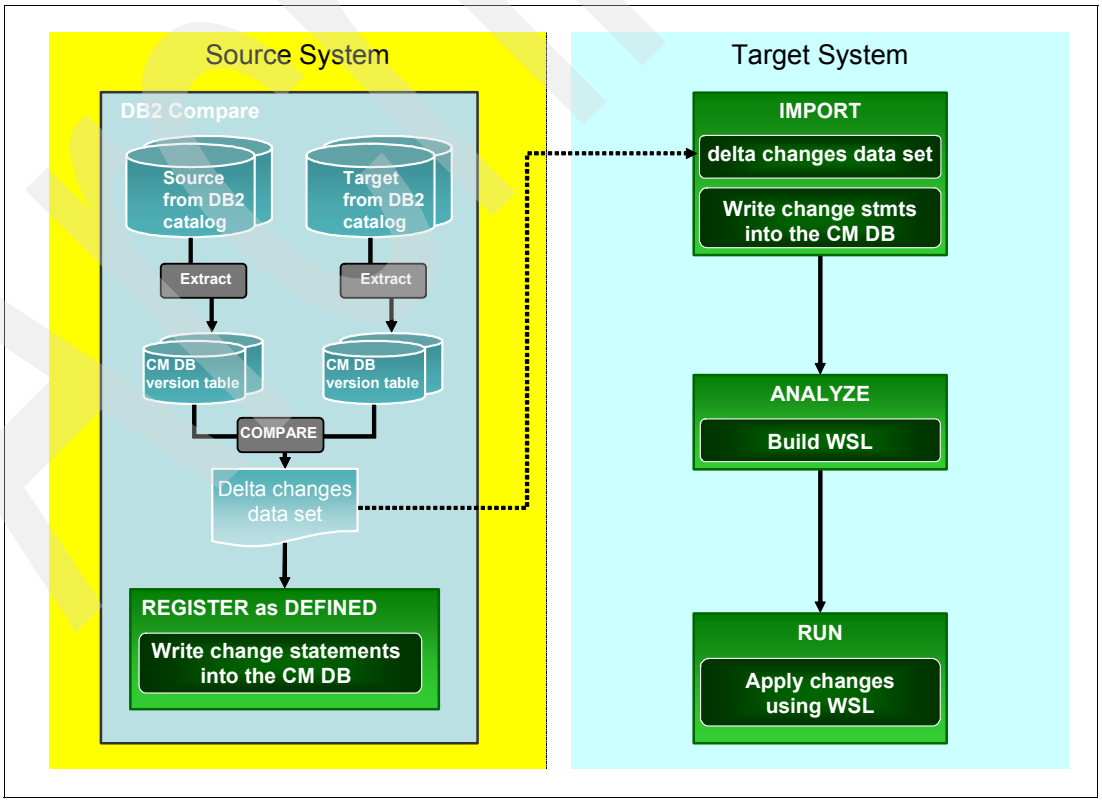

Figure 7-11 Rolling out using DB2 Compare

7.2.1 The change being made

With this example we compare a table DSG24D01.GLWTEPA against DSG24P01.GLWTEPA in the same subsystem. DSG24D01.GLWTEPA has a column UPDATED_BY defined with CHAR(80), DSG24P01.GLWTEPA has the same table definition but the column UPDATED_BY is defined with CHAR(8).

There is an additional index DSG24D01.GLWXEPA3 defined on table DSG24D01.GLWTEPA. This index does not exist on table DSG24P01.GLWTEPA.

7.2.2 Invoking DB2 Compare

After invoking DB2 Compare from the DB2 Admin Main Menu you have to specify the source and the target. According to our change, we use the table DSG24D01.GLWTEPA as a source and the table DSG24P01.GLWTEPA as a target. See Figure 7-12.

```
DB2 Admin -------------- DB2 Administration Menu 7.2.0 ------------------ 12:05
Option ===> C

   1 - DB2 system catalog                                DB2 System: DB8A
   2 - Execute SQL statements                            DB2 SQL ID: PAOLOR1
   3 - DB2 performance queries                           Userid    : PAOLOR1
   4 - Change current SQL ID                             DB2 Rel   : 815
   5 - Utility generation using LISTDEFs and TEMPLATEs
   P - Change DB2 Admin parameters
  DD - Distributed DB2 systems
   E - Explain
   Z - DB2 system administration
  SM - Space management functions
   W - Manage work statement lists
   X - Exit DB2 Admin
  CC - DB2 catalog copy version maintenance
  CM - Change management

Interface to other DB2 products and offerings:
   C  DB2 Object Comparison Tool
   I  DB2I
```

Figure 7-12 Invoking DB2 Compare

7.2.3 Specifying the compare source

To specify the source, we choose option **1**. See Figure 7-13.

```
Compare ---------------- DB2 Object Comparison Tool Menu ---------------- 13:10
Option ===> 1

                                                Status:
   1 - Specify compare source (new)             DB2 catalog extract specified
   2 - Specify compare target (old)             DB2 catalog extract specified
   3 - Specify compare masks                    Mask specified
   4 - Specify fields to ignore                 IGNORE Data Set Specified
   5 - Generate compare job                     Generated

   W - Walk through steps 1 - 5 in sequence
   V - Generate job to extract version file from source only
   R - Reset status

   S  - Save dialog
   M  - Manage/Restore dialog
   MC - MultiCompare
```

Figure 7-13 DB2 Compare Main Menu

We choose **2** to specify that source is from the DB2 catalog. See Figure 7-14.

```
Compare -------------------- Specify Compare Source --------------------- 13:11
Option ===> 2


   1 - Source is from a DDL file
   2 - Source is from the DB2 catalog
   3 - Source is from a compare version file
```

Figure 7-14 Specifying the source

Notice that you can specify the version as a version table entry. You have to specify either a version name as a version table entry or a version name as a data set name. The description is optional. See Figure 7-15.

Archived

```
Compare -------------- Specify DB2 Source Catalog Extract --------------- 13:12
Option ===> 3


   1 - Source is databases from the DB2 catalog
   2 - Source is table spaces from the DB2 catalog
   3 - Source is tables from the DB2 catalog
   4 - Add schema objects to the DB2 Source catalog extract

Specify compare version file output:
 Version table entry:
   Owner . . .  PAOLOR1  >              (? to look up)
   Name  . . .  D070321-CR#91-SRC  >    (? to look up)
 Data set:
  Data set name . .

Enter a description (optional):

  Description  . . SOURCE VERSION
```

Figure 7-15 The source is table

As a source name, we entered the table **DSG24D01.GLWTEPA**. See Figure 7-16.

```
Compare ----------------- Specify Source DB2 Tables -------------- Row 1 of 2
Command ===>                                              Scroll ===> CSR

Commands: RESET
Line commands:
 D - Delete  I - Insert

Select Owner    Table Name         Location
       *        *                  *
------ -------- -----------------  ---------------
*      ........ .................  ................
       DSG24D01 GLWTEPA            DB8A
****************************** END OF DB2 DATA ******************************
```

Figure 7-16 Specifying the table name

7.2.4 Specifying the compare target

In the same way as the compare source, you can specify the compare target. Just type option **2** and follow the panels you already know from specifying the compare source. Notice that we use the table **DSG24P01.GLWTEPA** as the compare target. See Figure 7-17.

```
Compare ---------------- DB2 Object Comparison Tool Menu ---------------- 13:41
Option ===> 2

                                                Status:
   1 - Specify compare source (new)             DB2 catalog extract specified
   2 - Specify compare target (old)             DB2 catalog extract specified
   3 - Specify compare masks                    Mask specified
   4 - Specify fields to ignore                 IGNORE Data Set Specified
   5 - Generate compare job                     Generated

   W - Walk through steps 1 - 5 in sequence
   V - Generate job to extract version file from source only
   R - Reset status

   S  - Save dialog
   M  - Manage/Restore dialog
   MC - MultiCompare
```

Figure 7-17 Specifying the compare target

7.2.5 Specifying compare masks

We have to make the naming identical before we start the compare process. For instance, if we did not mask the source table owner from DSG24D01 to DSG24P01 before comparing, then DB2 Compare would find that there is no counterpart for table DSG24D01.GLWTEPA in the target environment. Compare would then decide to generate a new table DSG24D01.GLWTEPA and all its dependencies in the target environment.

For that reason we mask the table owner before we do the Compare. DB2 Compare would then change the source table owner first before the compare process starts. In our example, the table owner DSG24D01 is changed to DSG24P01 first for the source table. So the compare process does not detect the table name as a difference and could focus on other differences between the two tables and its dependencies.

In our example, we choose a mask table entry rather than a masking data set. If you do not know exactly what your masking name is, put a **?** in the Owner and Name fields to look it up.

See Figure 7-18.

```
Compare --------------------- Specify Compare Masks ---------------------------
Option ===>


Mask Table Entry:
  Owner . .  PAOLOR1  >            (? to look up)
  Name  . .  MASTERMASK         >  (? to look up)
Data Set:
  Mask DSN  . .
Options:
  Edit Mask . .  YES (Yes/No)
```

Figure 7-18 Specifying a mask

We use PAOLOR1.MASTERMASK as our current mask entry. Because for Edit Mask we said **YES**, we see the current definition. See Figure 7-19.

```
DB2 Admin ----------------- CM - Mask Lines ----------------------- Row 1 of 1
Command ===>                                                   Scroll ===> CSR

Mask lines for mask "PAOLOR1"."MASTERMASK"
Commands: SAVE
Line commands:
  I - Insert  D - Delete  R - Repeat  M - Move  A - After  B - Before

Sel    Sequence Type     From                  To                    Oper.  T
              * *        *                     *                     *      *
--- ----------- -------- --------------------> --------------------> ------ -

              1 OWNER     DSG24D01             DSG24P01
****************************** END OF DB2 DATA ******************************
```

Figure 7-19 The current definition of mask PAOLOR1.MASTERMASK

As described, the owner from DSG24D01 to DSG24P01 changes before DB2 Compare starts comparing the source and target environments.

A detailed description of how to use masking is provided in 4.9, “Using the masking feature” on page 166.

7.2.6 Specifying fields to ignore

You can ignore some fields from comparing. Ignored fields are not extracted from the DB2 catalog when DB2 Compare starts. In other words, they are excluded from the compare process. In our example we do not use an ignore entry.

A detailed description of how to use ignore is provided in 4.10, “Using the ignore feature” on page 169.

7.2.7 Generating a compare job

When you enter the compare job parameters, make sure that you set the option GENERATE APPLY JOBS to YES. This is important because only then you are asked to use Change Management.

Notice that although you are asked to enter a WSL name, this WSL name is not used for creating a WSL member in a PDS. Furthermore, the WSL name is used as a middle qualifier for temporary files. See Figure 7-20.

```
Compare -------------------- Generate Compare Jobs  ---------------------------
Option ===>

 Specify the following for DB2 Object Comparison Tool:
                                                                   More:     +
 Worklist information:
   Worklist name . . . . . . . TEMPWSL  (also used as middle qualifier in DSNs)

 Compare options:
   Suppress DROP of objects  . YES      (Yes/No)
   Suppress DROP of columns  . NO       (Yes/No)
   Suppress adding columns . . NO       (Yes/No)
   Run SQLID . . . . . . . . . PAOLOR1  (Blank, an SQLID, or <NONE>)

 Change reporting options  . . NO       (Yes/No)

 Data set information:
   PDS for batch jobs  . . . . 'ADMIN.CM.CNTL'
   Prefix for data sets  . . . ADMIN

 Options:
   Single compare job  . . . . YES      (Yes/No)
     Member name . . . . . . . COMPARE  (default COMPARE )
   Generate apply jobs . . . . YES      (Yes/No)
     Generate one job  . . . . YES      (Yes, No, or (Per) Process)
       Member prefix . . . . . APPLY    (default APPLY )
     As work statement list  . NO       (Yes/No to append to work stmt list)
     Use utility options . . . NO       (Yes/No)
     Unload Method . . . . . . U        (Unload, Parallel unload)
     Generate templates. . . . NO       (Yes/No)
     Stop on conversion error. NO       (Yes/No)

     Optional jobs after Reload or Alter:
       Run CHECK DATA  . . . . NO       (Yes/No)
       Take an image copy  . . N        (after: Reload/Alter/Both/None)
       Run REORG/REBUILD . . . N        (Mandatory, All relevant, None)
       Run RUNSTATS  . . . . . N        (after: Reload/Alter/Both/None)
       Run REBIND  . . . . . . NO       (Yes/No)

 BP - Change batch job parameters
 TU - Specify TEMPLATE usage
 UO - Change utility options
```

Figure 7-20 The DB2 Compare parameter panel

Archived

You are then prompted if you want to use Change Management. See Figure 7-21.

```
DB2 Admin ----------- DB8A Change Management Prompt ----------- 15:16

Change Management is optional for SQLID:  PAOLOR1

Do you wish to use Change Management for this function: Y   (Yes/No)
```

Figure 7-21 Using Change Management

You are then asked to enter the change name and the change owner in the register panel. See Figure 7-22. You have to enter a name for the delta version as well, remember that a delta version name it is defaulted to the change owner and name.

```
DB2 Admin -------------------- CM - Register Options -------------------- 15:21
Option ===>

Commands: CONTINUE                                   DB2 System: DB8A
                                                     DB2 SQL ID: PAOLOR1
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR1  >  (Optional, default is PAOLOR1)
Name  . . . . . . . . D070316-CR#97-NOR                                  >
Comment . . . . . . .                                                    >
Change Type . . . . : CHANGE   (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                      Owner         Name
Ignore  . . . . . . .         >                          >  (? to look up)
Mask  . . . . . . . .         >                          >  (? to look up)
Delta Version . . . . PAOLOR1  >    D070316-CR#97-NOR    >  (? to look up)
```

Figure 7-22 Change registration with Compare

There is another option you have to specify after the registration panel. DB2 Compare has to know how you want to proceed with your change if pending changes are to be detected during the REGISTER process. The following options are available:

- Prereq:

  Allow pending changes to be prerequisites to these changes
- Cancel:

  If pending changes exist, do not register these changes
- Supersede:

  This change is to supersede pending changes

You are then asked to submit your compare job. Your change only becomes a registered change with a DEFINE status if your compare job ends successfully. Notice that there is a register step in the end of the Compare JCL which does the registration in Change Management. As the REGISTER process can be done exactly one time, you cannot run the same Compare JCL again. A second run of your compare job would abend with RC 16 in step T04REG, which is the registration step. See Figure 7-23.

```
 Display  Filter  View  Print  Options  Help
 ------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY ADBXXXXX JOB22745  DSID   132 LINE 25      COLUMNS 02- 81
 COMMAND INPUT ===>                                            SCROLL ===> CSR
      280 PAOLOR1  D070316-CR#96-NOR  CREATE INDEX DSG24D01.GLWXEPA3 ON DSG24D0
      280 PAOLOR1  D070316-CR#96-NOR  CREATE INDEX DSG24D01.GLWXEPA2 ON DSG24D0
      284 PAOLOR1  D070316-CR#97-NOR  CREATE INDEX DSG24P01.GLWXEPA3 ON DSG24P0
      284 PAOLOR1  D070316-CR#97-NOR  ALTER TABLE DSG24P01.GLWTEPA ALTER UPDATE


ADB2CID - End Pending Changes Report
Register error. rc=            16
ADBCRCM: Change already exists:PAOLOR1 D070316-CR#97-NOR
******************************** BOTTOM OF DATA ******************************
```

Figure 7-23 Error message if the DB2 Compare job is submitted twice

There is an important DD statement in the DB2 Compare JCL in Figure 7-24.

```
EDIT       ADMIN.CM.CNTL(COMPARE) - 01.00                Columns 00001 00072
Command ===>                                                Scroll ===> CSR
****** ***************************** Top of Data ******************************
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -   292 Line(s) not Displayed
000288 //CHANGES  DD DSN=ADMIN.TEMPWSL.CHANGES,
000289 //           DISP=(NEW,CATLG,DELETE),
000290 //           DCB=(LRECL=16384,RECFM=VB,BLKSIZE=27998),
000291 //           SPACE=(CYL,(10,10),RLSE),
000292 //           UNIT=SYSDA
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -   104 Line(s) not Displayed
****** **************************** Bottom of Data ****************************
```

Figure 7-24 The changes DD statement

The data set **ADMIN.TEMPWSL.CHANGES** is filled with the ‘delta’ between source and target. It contains the following DDL in Figure 7-25. This is the delta changes data set provided by DB2 Compare.

```
VIEW       ADMIN.TEMPWSL.CHANGES                          Columns 00001 00072
Command ===>                                                 Scroll ===> CSR
****** ***************************** Top of Data ****************************
000001 SQLID     SET CURRENT SQLID = 'PAOLOR1' ;
000002 CREATE    ALTER TABLE DSG24P01.GLWTEPA
000003 CREATE      ALTER UPDATED_BY SET DATA TYPE CHAR(80) ;
000004 CREATE    COMMIT ;
000005 CREATE    CREATE INDEX DSG24P01.GLWXEPA3
000006 CREATE      ON DSG24P01.GLWTEPA
000007 CREATE       (EMSTDATE              ASC,
000008 CREATE        EMENDATE              ASC)
000009 CREATE      USING STOGROUP GLWG01
000010 CREATE      PRIQTY 100 SECQTY 100
000011 CREATE      ERASE  NO
000012 CREATE      FREEPAGE 0 PCTFREE 5
000013 CREATE      GBPCACHE CHANGED
000014 CREATE      NOT CLUSTER
000015 CREATE      BUFFERPOOL BP16
000016 CREATE      CLOSE YES
000017 CREATE      COPY NO
000018 CREATE      DEFINE YES
000019 CREATE      PIECESIZE 2 G;
000020 CREATE    COMMIT ;
****** **************************** Bottom of Data ***************************
```

Figure 7-25 DB2 Compare delta changes data set

Notice that the resulting delta changes data set is NOT exactly the same as in the PROMOTE process. With the PROMOTE process, you compare two base versions that are based on the same database and generated with the same version scope. With Compare, we compared two tables that belong to different databases. In our Compare example, table DSG24D01.GLWTEPA has an index that is not defined in DSG24P01.GLWTEPA. Therefore, the delta changes data set from DB2 Compare shows the additional index.

You can look up the registered change in the Change Management Database. The change is in status DEFINED. See Figure 7-26.

```
DB2 Admin ------------------ CM - Changes ------------------------- Row 7 of 8
Command ===>                                                 Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- ---------------------
         284 PAOLOR1  D070316-CR#97-NOR  CHANGE  DEFINED
         285 PAOLOR1  D070316-CR#98-NOR  CHANGE  DEFINED
****************************** END OF DB2 DATA ******************************
```

Figure 7-26 The used change during the compare process

As your registered change contains valid statements now, you can then ANALYZE and RUN your change. If you did the compare just because you wanted to get the delta changes data set, then you can transfer the delta changes data set to another DB2 environment and proceed with importing the delta changes data set in the DB2 environment of your choice.

7.3 Importing delta changes data set

You can import your changes by just using your delta changes data set in the target environment of your choice. You can import a delta changes data set provided by DB2 Compare or provided by the PROMOTE process. See Figure 7-27.

```
DB2 Admin ------------------- CM - Import Changes ----------------------- 16:59
Command ===>


Input data set information:                              DB2 System: DB8A
 Data set name 'ADMIN.TEMPWSL.CHANGES'

Prereq resolution mode  . BATCH     (TSO/Batch)
Execution mode  . . . . . BATCH     (TSO/Batch)
```

Figure 7-27 Importing a delta change data set

You are then asked to register a change for this IMPORT process. In our example, we do directly a masking while importing the delta changes data set. We want to change every owner in our imported DDL from * (whatever) to DSG24P02. See Figure 7-28.

```
DB2 Admin -------------------- CM - Register Options -------------------- 17:07
Option ===>

Commands: CONTINUE                                       DB2 System: DB8A
                                                         DB2 SQL ID: PAOLOR1
Specify the following values to register a change:

Owner . . . . . . . PAOLOR1  >  (Optional, default is PAOLOR1)
Name  . . . . . . . D070316-CR#102-IMP                                   >
Comment . . . . . .                                                      >
Change Type . . . . : PROMOTE  (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                    Owner         Name
Ignore  . . . . . .         >                           >  (? to look up)
Mask  . . . . . . . PAOLOR1  >    PROD                  >  (? to look up)
Delta Version . . . . PAOLOR1  >    D070316-CR#102-IMP  >  (? to look up)
```

Figure 7-28 Register of your IMPORT

The generated batch job does the IMPORT for you. The change is in a DEFINED status. See Figure 7-29.

```
DB2 Admin ------------------ CM - Changes ----------------------- Row 11 of 11
Command ===>                                                  Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- ---------------------
ST       288 PAOLOR1  D070316-CR#102-IMP CHANGE  DEFINED
****************************** END OF DB2 DATA ******************************
```

Figure 7-29 Status of your imported change

When you look at the statements with **ST**, notice that the owner changed to DSG24P02 because of the masking. See Figure 7-30.

```
DB2 Admin ------------- CM - Change Statements -------------------- Row 1 of 2
Command ===>                                                  Scroll ===> CSR

Change statements for change "PAOLOR1"."D070316-CR#102-IMP"
Line commands:
 E - Edit  D - Delete  I - Insert  S - Show

Sel    Sequence O  Qual     Name                Statement
              * *  *        *                   *
--- ----------- -- -------- ----------------- ------------------------------>

             1 TB DSG24P02 GLWTEPA            ALTER TABLE DSG24P02.GLWTEPA ALT
             2 IX DSG24P02 GLWXEPA4           CREATE INDEX DSG24P02.GLWXEPA4 .
****************************** END OF DB2 DATA ******************************
```

Figure 7-30 Change statements of your change

Your change has now been imported in the DB2 target environment of your choice.

7.4 ANALYZE and RUN

You then can ANALYZE and RUN your change in the DB2 target environment where you have recently imported your change. Analyzing is a general checking process of validity. The Analyze process is described in detail in 4.4.3, “The ANALYZE process” on page 139. The RUN process finally applies your change to your desired target environment. The RUN process is described in detail in 4.4.4, “The RUN process” on page 145.

8

Recovery of changes

In this chapter we describe the process for recovering from a change. We explain the details of the RECOVER process by showing the steps required to accomplish the recovery.

We discuss five scenarios:

- Recovering a change with existing data (with RECOVER process)
- Recovering a change with original data (with RECOVER process)
- Recovering an uncompleted change to existing data (with generation of RECOVER WSL)
- Backing out to a previous version (without RECOVER process)
- Drop Recovery

© Copyright IBM Corp. 2007. All rights reserved.

8.1 Recovery process in detail

The recovery process follows the workflow diagram in Figure 8-1.

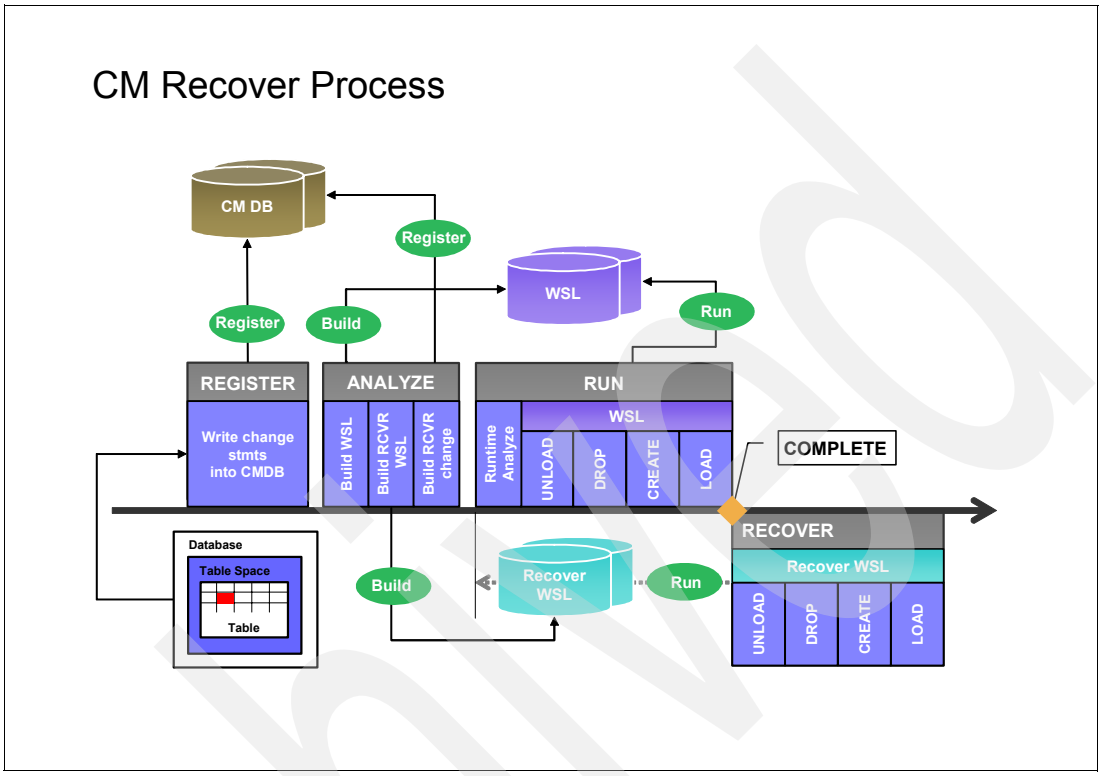

Figure 8-1 CM Recover Process in detail

8.2 Recovery considerations for a completed change

Before we start with implementing changes through CM, we should make a snapshot of the structure — a base version — before the first change is going to happen. This is the first base version of our structure (often referred to as V0), and it represents the structure before the change. This V0 then can be used as starting point for later comparisons with other versions (V1, V2, V3, Vn).

We have to generate this first base version ourselves, so it is good practice to do this up front. Version files (including base versions) are explained in detail in 4.12, “Versions” on page 186. This manual generation of a base version only has to be done once; at all other times it can be done automatically through the product. When a change has been done and the according option has been selected, then after the RUN — and the change has status COMPLETE — the job also takes a base version (V1).

So here is the flow of activities we want to show:

- Take a base version (V0)
- Do the first change against a table
- Take a base version (V1)
- Run the recover process to back out the first change

When you decide to generate a RECOVER WSL, then you also have to decide with what data you want to recover back:

- Existing data
- Original data

It is not always apparent what decision to make at ANALYZE time, because you might not exactly know with what data (current or original) you have to go back in case of a RECOVER. And also, depending on the change, there is not always the possibility that the recover WSL can be generated with original data (see also Table 4-4 on page 159).

However, it is good practice to generate a recover WSL. It gives you always the option to recover later and always carries the DDL in the recover WSL. The question is: What is the better option: ORIGINAL or EXISTING data?

If you make a base version after every change, then you can always reverse a change back by comparing these base versions (compare V1 against V0, or compare V3 against V1). This strategy enables you to be able to recover a change with existing data — whether it is through the recover process or through a normal change process.

This means that if you default to ORIGINAL data, then there is always an option to get back with existing data by comparing the current DB2 catalog with a previously done base version.

The other benefit when working with ORIGINAL data as a RECOVER default is that, when an object gets dropped, you always have the CREATE DDL and, depending on the change type, the LOAD statement in the RECOVER WSL. This helps in normal situations where a table gets dropped and you want to retain the ability to get back to it with data.

On the other hand, if you default RECOVER to EXISTING data, then to get back with the old data would require some manual intervention (you would have to manipulate the WSL and refer to the old unload file, providing that it is still around).

Tips:

- Always take base versions after a change.
- Consider always using **`RECOVER: YES`**.

If changes have occurred, then you might ask the question: Does the change back cause any data loss due to conversion issues? This is possible if the column type changes to a larger size (like from char10 to char20) and now the reverse change goes from char20 to char10. But what happens if there was already data changed on that column? A reversal truncates the current data.

Or when a column was dropped and later you want to go back to the old structure, then existing data does not help because it does not have this column. In this case *only* original data can help.

Another situation is when a column was added and data got inserted. The column gets dropped again when the change gets recovered.

So you have to keep such situations in mind when deciding to recover.

8.2.1 Recovering a completed change with existing data

This scenario describes step by step what has to be done to recover a completed change made through change management. We use the RECOVER process to revert a change using existing data (see Figure 8-2).

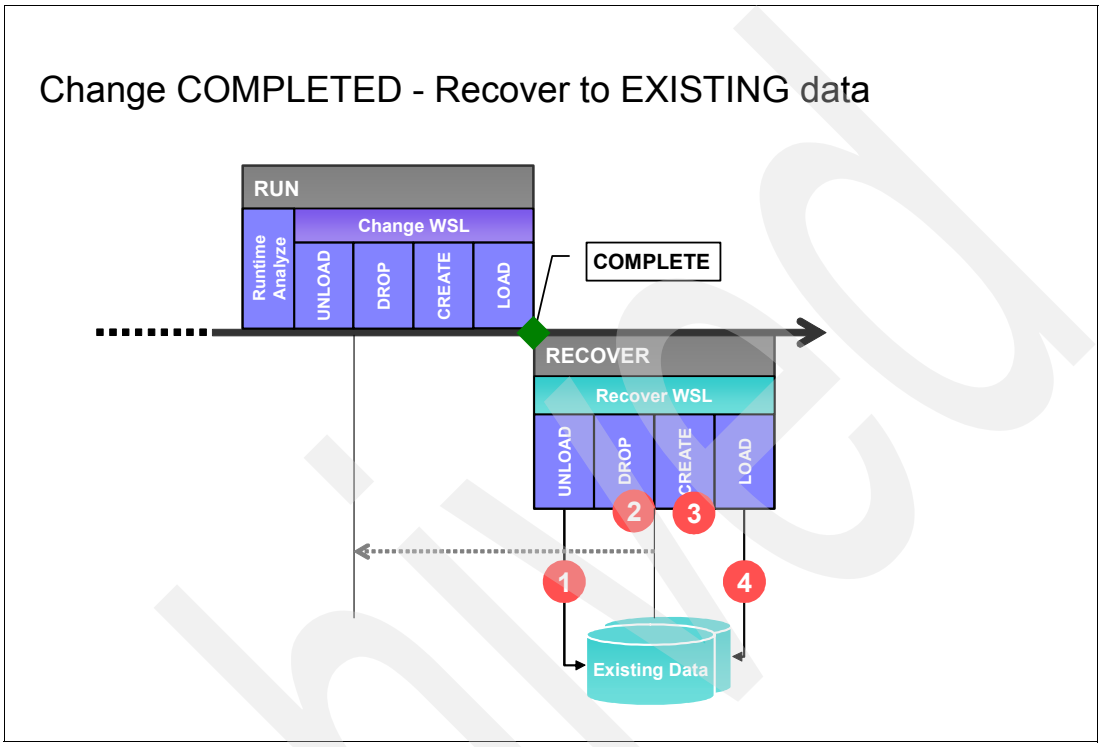

Figure 8-2 Recovery of a completed change with existing data

The following actions are done during Recover WSL execution:

1. Unload existing data
2. Drop created objects
3. Re-create the objects with the previous structure
4. Re-load existing data

We do three changes to one of our tables:

- Add column NEWCOL to the end of the table
- Drop column OLDCOL in the table
- Change view to add NEWCOL and drop OLDCOL

Reverting that change looks like this:

- Drop column NEWCOL
- Add column OLD (back again)
- Change view to drop NEWCOL and add OLDCOL

Now because we are working with current data, basically we lose what was in V0 for column OLD. We go back to the structure V0 but the data that was at that time in OLD is gone because it is not part of V1. Dropping column NEWCOL (and the data of it) should not hurt so much, because we decided to take off the column anyway. So this already gives us a hint that when we drop columns (or other objects), then we can better prepare a RECOVER with ORIGINAL data.

Archived

So we start from the beginning and we first take a base version of our database DSG24R02. Before we can do this we have to build a version scope — which defines what is to be stored in our base versions. How this is accomplished can be seen in “Creating a version scope” on page 188. So we check whether the scope has been defined to suit our naming conventions. See Figure 8-3.

We specify **SO** to get an overview of the objects that are included in the scope called PAOLOR2.DSG24R02. We named the scope after the database name to better relate what is contained in the scope.

```
DB2 Admin --------------- CM - Version Scopes --------------------- Row 1 of 5
Command ===>                                                  Scroll ===> PAGE

Line commands:
 VE - Versions  SO - Version scope objects  GV - Generate new base version
 INS - Insert  U - Update  DEL - Delete  I - Details on version scope

Sel          ID Owner    Name                Comment
              * *        *                   *
--- ----------- -------- ------------------ ---------------------------------
              1 PAOLOR2  PAOLOR2Z            SCOPE IS DATABASE
              3 PAOLOR3  DSG24D0%            SCOPE ALL DATABASES DSG24D0%
SO           21 PAOLOR2  DSG24R02            SCOPE IS DATABASE
             22 PAOLOR1  D070309-CR#33-SCP
             24 PAOLOR2  DSG24RZZ SCOPE      SCOPE IS ON DB LEVEL FOR DSG24RZZ
****************************** END OF DB2 DATA ******************************
```

Figure 8-3 Version Scopes panel

In our case we see that object type DB (for database) with name DSG24R02 is our scope. That means all objects in that database (table space, table, index, etc.) are included when a base version is created. See Figure 8-4.

```
DB2 Admin ----------- CM - Version Scope Objects ------------------ Row 1 of 1
Command ===>                                                  Scroll ===> PAGE

Version scope objects for scope "PAOLOR2"."DSG24R02"
Commands: SAVE
Line commands:
  I - Insert  D - Delete  R - Repeat

Sel T  Qual     Name                Oper.
    *  *        *                   *
--- -- -------> -----------------> ------

    DB          DSG24R02
****************************** END OF DB2 DATA ******************************
```

Figure 8-4 Version Scope Objects

In our scenario we work on the database level (object type DB) and in this case there is no qualifier required. For this reason we left this blank. Then we back out with PF3 and issue VE against the version scope to get a list of all available base versions. The result can be seen in Figure 8-5.

```
DB2 Admin ------------------ CM - Versions ------------------------ Row 1 of 1
Command ===>                                                Scroll ===> PAGE

Line commands:
 CH - Changes  PR - Promote  VS - Version scope  DEL - Delete  U - Update
 PT - Toggle protected status  I - Details on version

Sel          ID T Owner    Name                Comment
              * * *        *                   *
--- ----------- - -------- ----------------- -------------------------------
            138 B PAOLOR2  DSG24RO2 V0
******************************** END OF DB2 DATA ******************************
```

Figure 8-5 Versions panel

So when we have one V0 base version for database **DSG24RO2**, we can start to make the first changes. Then we go ahead and pull up the table we want to change in DB2 Admin and issue **ALC** against it. See Figure 8-6.

```
DB2 Admin ------------------- DB8A ALTER Table ------- Warning] ROWID/IDENTITY
Command ===>                                                Scroll ===> PAGE

New owner ===> DSG24RO2 >                   Old owner : DSG24RO2
New name  ===> GLWTSFN           >          Old name  : GLWTSFN
Volatile  ===> NO        Action    : NONE   New DB ===> DSG24RO2
Rows/Page    : 107.78    Partitions: 0      New TS ===> GLWS001
Commands :  CONTINUE  PRIMKEY  ADD  REL
Line commands :
 I - Insert  U - Update  D - Delete  R - Repeat  LAB - Label  COM - Comment
 M - Move    A - After   B - Before  X - Index   RES - Reset update
                                                            Old Operation
Select Column Name        Col No Col Type Length  Scale Null D Col No Type
       *                   * *            *      * *    *      * *
------ -----------------> ------ -------- ------ ------ ---- - ------ ---------

       SFN_NO                1 SMALLINT      2      0 N    J      1
       SEX                   2 CHAR          1      0 N    Y      2
       FIRSTNAME             3 VARCHAR      12      0 N    Y      3
******************************** END OF DB2 DATA ******************************
```

Figure 8-6 ALTER table with warning in upper right

First of all, in the upper right corner, we see a warning that our table contains a ROWID and/or IDENTITY column that is defined as GENERATED ALWAYS (A or I in column D on this panel). DB2 Admin Alter changes the definition of any ROWID or IDENTITY columns to GENERATED BY DEFAULT (D OR J) if the table is dropped and recreated in order to preserve the data values for that column when the table is reloaded. Any request to change the GENERATED definition is not honored to prevent the destruction of data during an unload/reload sequence of operations.

Tip: If you want IDENTITY columns in a table and there is ever a risk of altering the structure of that table, do not define it as GENERATED ALWAYS. You might consider using SEQUENCEs instead.

Our change is as follows:

- Drop the column SEX from the table.
- Insert a new column SECONDNAME (at the end).
- Update the view DDL.

We do not document the panel flow with ALC — we show the panels beginning from the registering point.

When we have done all the definitions, we then get to register the change as shown in Figure 8-7.

```
DB2 Admin -------------------- CM - Register Options -------------------- 18:17
Option ===>

Commands: CONTINUE                                      DB2 System: DB8A
                                                        DB2 SQL ID: PAOLOR2
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR2  >  (Optional, default is PAOLOR2)
Name  . . . . . . . . CH91-01                                            >
Comment . . . . . . . 1st change chapter 9.1 with current data           >
Change Type . . . . : CHANGE   (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                     Owner         Name
Ignore  . . . . . . .          >                          >  (? to look up)
Mask  . . . . . . . .          >                          >  (? to look up)
Delta Version . . . .          >                          >  (? to look up)
```

Figure 8-7 Register Options

We have provided a meaningful name and comment and then registered the change. See Figure 8-8.

```
DB2 Admin -------------------- CM - Register Options -------------------- 18:19
Option ===>
Register Successful
Commands: CONTINUE                                      DB2 System: DB8A
                                                        DB2 SQL ID: PAOLOR2
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR2  >  (Optional, default is PAOLOR2)
Name  . . . . . . . . CH91-01                                            >
Comment . . . . . . . 1ST CHANGE CHAPTER 9.1 WITH CURRENT DATA           >
Change Type . . . . : CHANGE   (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                     Owner         Name
Ignore  . . . . . . .          >                          >  (? to look up)
Mask  . . . . . . . .          >                          >  (? to look up)
Delta Version . . . . PAOLOR2  >    CH91-01               >  (? to look up)
```

Figure 8-8 Register Options panel with successful register message

Archived

Our previous ALC activity and the register process have written the change statements into the CM database. We then want to ANALYZE the change. For doing this we issue **CMM** on the command line to get to the CM main menu. We drill down to the changes and have a look at the change statements first (Figure 8-9) in order to see what has actually been written.

```
DB2 Admin ------------------ CM - Changes ------------------------ Row 1 of 79
Command ===>                                                  Scroll ===> PAGE

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
ST       155 PAOLOR2  CH91-01             CHANGE  DEFINED  1ST CHANGE CHAPTER 9.
         154 PAOLOR2  P024                COMPARE COMPLETE PROMOTE 3RD CHANGE RZ
         153 PAOLOR2  C024                CHANGE  COMPLETE 3RD CHANGE RZZ
         152 PAOLOR2  P023                COMPARE COMPLETE PROMOTE 2ND CHANGE RZ
         151 PAOLOR2  C023                CHANGE  COMPLETE 2ND CHANGE RZZ
```

Figure 8-9 Changes panel

These are the generated change statements from our previous ALC activity (see Figure 8-10).

```
DB2 Admin ------------- CM - Change Statements -------------------- Row 1 of 4
Command ===>                                                  Scroll ===> CSR

Change statements for change "PAOLOR2"."CH91-01"
Line commands:
 E - Edit  D - Delete  I - Insert  S - Show

Sel    Sequence O  Qual     Name                Statement
             * *  *        *                   *
--- ----------- -- -------- ------------------ ------------------------------>

             1 TB DSG24RO2 GLWTSFN             ALTER TABLE "DSG24RO2"."GLWTSFN
             2 TB DSG24RO2 GLWTSFN             ADMIN ALTER TABLE "DSG24RO2"."G
             3 VW DSG24RO2 WLSEXFIRSTNAME      ADMIN DROP VIEW  "DSG24RO2"."WL
             4 VW DSG24RO2 WLSEXFIRSTNAME       CREATE VIEW DSG24RO2.WLSEXFIRS
****************************** END OF DB2 DATA ******************************
```

Figure 8-10 Change Statements panel

We confirm briefly that these statements are what we expected to see and move on.

Tip: If you select each single line with **S** (Show) then you can see the details for every change:

```
ALTER TABLE "DSG24RO2"."GLWTSFN"   ADD "SECONDNAME"   VARCHAR  (12) FOR
SBCS DATA NOT NULL WITH DEFAULT

ADMIN ALTER TABLE "DSG24RO2"."GLWTSFN"    DROP COLUMN "SEX"

ADMIN DROP VIEW  "DSG24RO2"."WLSEXFIRSTNAME"

CREATE VIEW DSG24RO2.WLSEXFIRSTNAME(SFN_NO, FIRSTNAME, SECONDNAME) AS
  SELECT SFN_NO, FIRSTNAME, SECONDNAME
    FROM DSG24RO2.GLWTSFN T
```

We press PF3 to get back to the changes overview and enter the **AN** command against the change ID we want to analyze. See Figure 8-11.

```
DB2 Admin ------------------ CM - Changes ------------------------ Row 1 of 79
Command ===>                                                  Scroll ===> PAGE

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                 Type    Status   Comment
           * *        *                    *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
AN       155 PAOLOR2  CH91-01              CHANGE  DEFINED  1ST CHANGE CHAPTER 9.
         154 PAOLOR2  PO24                 COMPARE COMPLETE PROMOTE 3RD CHANGE RZ
         153 PAOLOR2  CO24                 CHANGE  COMPLETE 3RD CHANGE RZZ
         152 PAOLOR2  PO23                 COMPARE COMPLETE PROMOTE 2ND CHANGE RZ
         151 PAOLOR2  CO23                 CHANGE  COMPLETE 2ND CHANGE RZZ
```

Figure 8-11 Changes panel

Then we get to the ANALYZE panel (Figure 8-12).

```
DB2 Admin ---------------- Generate Analyze Job  ---------------------- 18:22
Command ===>

 Specify the following for Analyze:
                                                                  More:   - +
 Change reporting options  . . NO       (Yes/No)

 Required data set information:
   PDS for WSL . . . . . . . 'ADMIN.CM.WSL'
   PDS for batch jobs  . . . 'ADMIN.CM.JCL'
   Prefix for data sets  . . PAOLOR2

 Options:
   Run SQLID . . . . . . . . <NONE>   (Blank, a SQLID, or <NONE>)
   Validate WSL  . . . . . . YES      (Yes/No)
   Use utility options . . . NO       (Yes/No)
   Generate templates  . . . NO       (Yes/No)
   Build JCL to run WSL  . . NO       (Yes/No)
   Generate a recover change YES      (Yes/No)
     Data to recover . . . . E        (Original or Existing)
```

Figure 8-12 Generate Analyze Job panel

Here we provide the settings for the generation of the RECOVER process. In this scenario we decided to work with **EXISTING** data for the RECOVER process. Since we specified **YES** for the field, Generate a recover change, we get prompted to provide information for the register process so that the RECOVER itself can be registered (see Figure 8-13).

```
DB2 Admin -------------------- CM - Register Options -------------------- 18:25
Option ===>

Commands: CONTINUE                                  DB2 System: DB8A
                                                    DB2 SQL ID: PAOLOR2
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR2  >  (Optional, default is PAOLOR2)
Name  . . . . . . . . RC91-01                                          >
Comment . . . . . . . RECOVER 1st CHANGE for DSG24RO2                  >
Change Type . . . . : RECOVER  (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                     Owner         Name
Ignore  . . . . . . .         >                          >  (? to look up)
Mask  . . . . . . . .         >                          >  (? to look up)
Delta Version . . . .         >                          >  (? to look up)
```

Figure 8-13 Register Options panel for RECOVER

We provide a meaningful name and comment so that later we can easily relate that name to the change it belongs to (our change name is CH91-01 and we specified RC91-01 for the recover change name). Then we CONTINUE and get the analysis job generated (see Figure 8-14).

```
EDIT       ADMIN.CM.JCL(C0000155) - 01.00                 Columns 00001 00072
Command ===>                                                 Scroll ===> CSR
****** **************************** Top of Data ****************************
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  7 Line(s) not Displayed
000008 //*******************************************************************
000009 //* DB2 ADMINISTRATION TOOL GENERATED JOB
000010 //*
000011 //* THIS IS A CHANGE MANAGEMENT ANALYZE JOB FOR CHANGE ID 155.
000012 //*
000013 //* FOR RESTART, RESUBMIT THE ENTIRE JOB.
000014 //*
000015 //*************************************************************ADB2C11A**
000016 //        IF (RC>7) THEN
000017 //        ELSE
000018 //*******************************************************************
000019 //*
000020 //* STEP EXDLTA: EXTRACT ALL RELATED DELTA VERSION FILES FROM DB2
000021 //*
000022 //*********************************************************ADBCVEX*********
000023 //EXDLTA   EXEC PGM=IKJEFT01,DYNAMNBR=100
000024 //STEPLIB  DD DISP=SHR,DSN=GOC.V7R2M0.SGOCLLIB
```

Figure 8-14 Generated analysis Job

This job can then be submitted. If everything runs OK, we have two changes in **ANALYZED** status, as can be seen in Figure 8-15.

```
DB2 Admin ------------------ CM - Changes ------------------------ Row 1 of 80
Command ===>                                                 Scroll ===> PAGE

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
         156 PAOLOR2  RC91-01            RECOVER ANALYZED RECOVER 1ST CHANGE FO
         155 PAOLOR2  CH91-01            CHANGE  ANALYZED 1ST CHANGE CHAPTER 9.
         154 PAOLOR2  P024               COMPARE COMPLETE PROMOTE 3RD CHANGE RZ
         153 PAOLOR2  C024               CHANGE  COMPLETE 3RD CHANGE RZZ
         152 PAOLOR2  P023               COMPARE COMPLETE PROMOTE 2ND CHANGE RZ
```

Figure 8-15 Changes panel showing ANALYZED status for RECOVER and CHANGE type

Archived

Now we have actually two work statement lists generated and we want to show where they are located. Since the change ID is 155 (type CHANGE) we can find the WSLs in members C0000155 (for the change) and R0000155 (for the recover) in the WSL data set we specified on the ANALYZE panel, which was in our case **ADMIN.CM.WSL**. See Figure 8-16.

```
DB2 Admin ----------- Work Statement List Library: 'ADMIN.CM.WSL' Row 29 of 33
Command ===>                                                  Scroll ===> PAGE

Line commands:
 S - Show  R - Run in batch  D - Delete  C - Copy  A - Append  Q - Clone
 I - Interpret  V - Validate  E - Edit

Sel Name      Size Created    Changed          ID
    *            * *          *                *
--- -------- ----- ---------- ---------------- --------
    C0000155    90 2007/03/12 2007/03/12 18:28 PAOLOR2
    R0000094    91 2007/03/08 2007/03/08 16:54 PAOLOR2
    R0000096    91 2007/03/08 2007/03/08 17:23 PAOLOR2
    R0000111  1211 2007/03/09 2007/03/09 21:25 PAOLOR2
    R0000155    91 2007/03/12 2007/03/12 18:29 PAOLOR2
****************************** END OF DB2 DATA ******************************
```

Figure 8-16 Work Statement List Library panel

You might notice that the recover change ID is 156 — but its actual work statement list (R0000155) carries the change ID — and not the change ID of the recover. So when we are back on the changes panel, we can RUN the change (see Figure 8-17).

```
DB2 Admin ------------------ CM - Changes ------------------------ Row 1 of 80
Command ===>                                                  Scroll ===> PAGE

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                 Type    Status   Comment
           * *        *                    *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
         156 PAOLOR2  RC91-01              RECOVER ANALYZED RECOVER 1ST CHANGE FO
RN       155 PAOLOR2  CH91-01              CHANGE  ANALYZED 1ST CHANGE CHAPTER 9.
         154 PAOLOR2  P024                 COMPARE COMPLETE PROMOTE 3RD CHANGE RZ
         153 PAOLOR2  C024                 CHANGE  COMPLETE 3RD CHANGE RZZ
```

Figure 8-17 Changes panel

Archived

We enter **RN** against change ID 155 and then we get to the panel where we can specify the RUN parameters (see Figure 8-18).

```
DB2 Admin ---------------- CM - Run a Change ----------------------------------
Command ===>
Enter required field

Change . . . : PAOLOR2.CH91-01


Data set information:
  PDS for batch jobs . . . . . 'ADMIN.CM.JCL'
  Prefix for data sets . . . . PAOLOR2

Create a new base version  . . YES  (Yes/No)

  Scope Information:
    Owner  . . . . . . . . . . PAOLOR2 >          (? to look up)
    Name . . . . . . . . . . . DSG24R02          >  (? to look up)

  Version Information:
    Owner  . . . . . . . . . . PAOLOR2 >          (? to look up)
    Name . . . . . . . . . . . DSG24R02 v1       >  (? to look up)

Runtime analyze:
  Change reporting options . . NO   (Yes/No)
```

Figure 8-18 Run a change panel

On this panel we specify that we also want to create a new base version and specify **YES** to it. With this we also get JCL generated that creates a new base version (in our case V1) if the RUN of the change completes. We could also skip this new base version generation on this panel, but that would require us to remember it later and take that base version (V1) anyway. So it is good practice to have this generated along with the RUN JCL.

One thing to remember is that the version name (in our case **DSG24R02 V1**) should be carefully chosen. Best practice would be to follow a logical sequence so that later we can pull up a list of all versions and easily determine the sequence of versions.

Archived

After we have specified all that, we press Enter and get the generated job (see Figure 8-19).

```
EDIT       ADMIN.CM.JCL(E0000155) - 01.00                 Columns 00001 00072
Command ===>                                                 Scroll ===> CSR
****** **************************** Top of Data ******************************
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  7 Line(s) not Displayed
000008 //*********************************************************************
000009 //* DB2 ADMINISTRATION TOOL GENERATED JOB
000010 //*
000011 //* THIS IS A CHANGE MANAGEMENT RUN JOB FOR CHANGE ID 155.
000012 //*
000013 //* FOR RESTART:
000014 //*   IF THE JOB FAILS ON OR AFTER THE STEP TO RUN THE ADBTEP2 PROGRAM
000015 //*   (E.G. WITH STEP NAME BEGINNING WITH DB2B), RESUBMIT AT THE FAILING
000016 //*   STEP.  OTHERWISE, RESUBMIT THE ENTIRE JOB.
000017 //*
000018 //*************************************************************ADB2C11A**
```

Figure 8-19 Generated Job for the change

Once we are ready we can submit the job and get the changes done. If the changes were successful, we then get the confirmation on the changes panel. Change ID 155 is now in COMPLETE status. Also notice that the RECOVER type stays in ANALYZED status (see Figure 8-20).

```
DB2 Admin ------------------ CM - Changes ------------------------ Row 1 of 80
Command ===>                                                 Scroll ===> PAGE

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                 Type    Status   Comment
           * *        *                    *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
         156 PAOLOR2  RC91-01              RECOVER ANALYZED RECOVER 1ST CHANGE FO
         155 PAOLOR2  CH91-01              CHANGE  COMPLETE 1ST CHANGE CHAPTER 9.
         154 PAOLOR2  PO24                 COMPARE COMPLETE PROMOTE 3RD CHANGE RZ
         153 PAOLOR2  CO24                 CHANGE  COMPLETE 3RD CHANGE RZZ
```

Figure 8-20 CM Changes panel showing the change in COMPLETE status

So we are now on V1 of the database DSG24R02. We should also have a second base version now that reflects the new structure.

If for any reason you might want to go back to V0, then you can take the chance and use the RECOVER functionality within change management. Since we have specified RECOVER when we did the first change, we can actually use it now. See Figure 8-21.

```
DB2 Admin ------------------ CM - Changes ------------------------ Row 1 of 80
Command ===>                                                  Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
         156 PAOLOR2  RC91-01            RECOVER ANALYZED RECOVER 1ST CHANGE FO
         155 PAOLOR2  CH91-01            CHANGE  COMPLETE 1ST CHANGE CHAPTER 9.
         154 PAOLOR2  PO24               COMPARE COMPLETE PROMOTE 3RD CHANGE RZ
```

Figure 8-21 Status of changes

If we want to recover, we have no other choice than to issue **RC** (for recover) against the change ID that was implementing the change (change ID 155). See also Figure 8-22.

Important: You cannot issue the **RC** command against the RECOVER type change ID.

```
DB2 Admin ------------------ CM - Changes ------------------------ Row 1 of 80
Command ===>                                                  Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
         156 PAOLOR2  RC91-01            RECOVER ANALYZED RECOVER 1ST CHANGE FO
RC       155 PAOLOR2  CH91-01            CHANGE  COMPLETE 1ST CHANGE CHAPTER 9.
         154 PAOLOR2  PO24               COMPARE COMPLETE PROMOTE 3RD CHANGE RZ
         153 PAOLOR2  CO24               CHANGE  COMPLETE 3RD CHANGE RZZ
```

Figure 8-22 CM Changes panel and issuing the RC (recover) command

Archived

Specifying RC then checks whether there are any pending changes which are affected by the RECOVER process. If there are pending changes, as in our case, we get another panel which tells you that he has to change the PCACT parameter in order to proceed with the recover execution (see also Figure 8-23).

```
DB2 Admin ---------------- DB8A CM - Recover Strategy ------------- Row 1 of 2
Command ===>                                                  Scroll ===> CSR

Commands: CONTINUE
Recover strategy for change "PAOLOR2"."CH91-01"
Line commands:
 CH - Change  I - Interpret

    Rcvr
Sel Order Owner    Name                Statement
        * *        *                   *
--- ----- -------- ------------------ ------------------------------------->
        0 PAOLOR3  RECOVERY TEST        ADMIN ALTER TABLE "DSG24D04"."GLWTVRN"
        0 PAOLOR3  RECOVERY TEST       ADMIN ALTER TABLE "DSG24D04"."GLWTVRN"
****************************** END OF DB2 DATA ******************************
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
 These pending changes need to be superseded in order for the change to be    e
 recovered.  Each of these changes that are not in DEFINED status will be set e
 to DEFINED.  You must ensure the PCACT parameter in the recover job is set   e
 to supersede (e.g. PCACT(SUPERSEDE)) to confirm the supersede action.        e
 Note: The recover strategy is re-calculated at runtime and thus may be       e
 different from what it is now.                                               e
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

Figure 8-23 Recovery Strategy panel showing pending changes

Attention: The recover strategy is re-calculated at runtime and thus can be different from what it is now.

This does not mean that the recover work statement list gets rebuilt !

At runtime, the recover strategy is checked again. The change is not recovered if the recover strategy at runtime determines that the change cannot be recovered at that time.

You have to find in the job the ADBTEP2 program that uses the PCACT PARM. By default, this value is generated with PCACT(CANCEL), and this means that when there any pending changes at the time of the worklist execution, then the RECOVER WSL terminates. See Figure 8-24.

```
****** ***************************** Top of Data ******************************
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  8 Line(s) not Displayed
000009 //* DB2 ADMINISTRATION TOOL GENERATED JOB
000010 //*
000011 //* THIS IS A CHANGE MANAGEMENT RECOVER JOB FOR CHANGE ID 155.
000012 //*
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 34 Line(s) not Displayed
000047 //SYSTSIN  DD *
000048  DSN SYSTEM(DB8A)
000049  RUN  PROGRAM(ADBTEP2) PLAN(A72) -
000050    LIB('ADB.V7R2M0.SADBLLIB')  -
000051    PARMS('/WORKLIST(R0000155) SSID(DB8A)  -
000052  RESTART(YES),RECOVER(155),PCACT(CANCEL)')
000053  END
000054 //SYSIN    DD *
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -   133 Line(s) not Displayed
****** **************************** Bottom of Data ****************************
```

Figure 8-24 Generated recover JCL with PCACT PARM

You have to manually change this **PCACT(CANCEL)** PARM to **PCACT(SUPERSEDE)** in order to start the recover WSL. When this change has been done, you can submit the job and see the RECOVER change on the changes panel in RUNNING status (Figure 8-25).

```
DB2 Admin ----------------- CM - Changes ------------------------ Row 1 of 80
Command ===>                                                  Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
        156 PAOLOR2  RC91-01             RECOVER RUNNING  RECOVER 1ST CHANGE FO
        155 PAOLOR2  CH91-01             CHANGE  DEFINED  1ST CHANGE CHAPTER 9.
        154 PAOLOR2  P024                COMPARE COMPLETE PROMOTE 3RD CHANGE RZ
        153 PAOLOR2  C024                CHANGE  COMPLETE 3RD CHANGE RZZ
```

Figure 8-25 Changes panel showing recover in running status

Notice that the change CH91-01 went into DEFINED status. When the recover WSL has finished successfully, then we see the RECOVER in COMPLETE status (Figure 8-26).

```
DB2 Admin ------------------ CM - Changes ------------------------ Row 1 of 80
Command ===>                                                  Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
         156 PAOLOR2  RC91-01            RECOVER COMPLETE RECOVER 1ST CHANGE FO
         155 PAOLOR2  CH91-01            CHANGE  DEFINED  1ST CHANGE CHAPTER 9.
         154 PAOLOR2  PO24               COMPARE COMPLETE PROMOTE 3RD CHANGE RZ
         153 PAOLOR2  CO24               CHANGE  COMPLETE 3RD CHANGE RZZ
```

Figure 8-26 Changes panel showing recover in complete status

Now we have put the DDL structure back in place as it was in V0 (see Figure 8-27).

```
DB2 Admin -- DB8A Columns in Table DSG24RO2.GLWTSFN    ------------ Row 1 of 3
Command ===>                                                  Scroll ===> PAGE

Line commands:
 T - Tables  X - Indexes  A - Auth  GR - Grant  H - Homonyms  I - Interpret
 UR - Update runstats  LAB - Label  COM - Comment  DI - Distribution stats
 PST - Partition stats   E  - Source data type  SEQ - Identity column info

Select Column Name        Col No Col Type Length Scale  Null Def FP    Col Card
       *                   * *            *      * *    *   *            *
------ ------------------ ------ -------- ------ ------ ---- --- -- -----------
       SFN_NO                  1 SMALLINT      2      0 N    J   N           84
       SEX                     2 CHAR          1      0 N    Y   N            1
       FIRSTNAME               3 VARCHAR      12      0 N    Y   N           83
****************************** END OF DB2 DATA ******************************
```

Figure 8-27 Columns of recovered table

The only difference though is that we probably might have other data now as it was at the time when we changed V0 to V1. This is because we dropped a column in the change and re-instated the column but we were using existing data to recover — which does not contain this re-instated column.

8.2.2 Recovering a completed change with original data

This scenario is an exact duplicate to the previous one — with the only exception that when we select the ANALYZE options, we choose for RECOVER the option **0** to work with original data (see Figure 8-28).

The following actions are done during Recover WSL execution:

1. Unload original data
2. Drop created objects
3. Re-create the objects with the previous structure

4. Re-load original data:

We do not show all steps in detail because the previous scenario shows these steps. Here, we provide additional information which helps you to better understand what happens behind the scene to recover the original data.

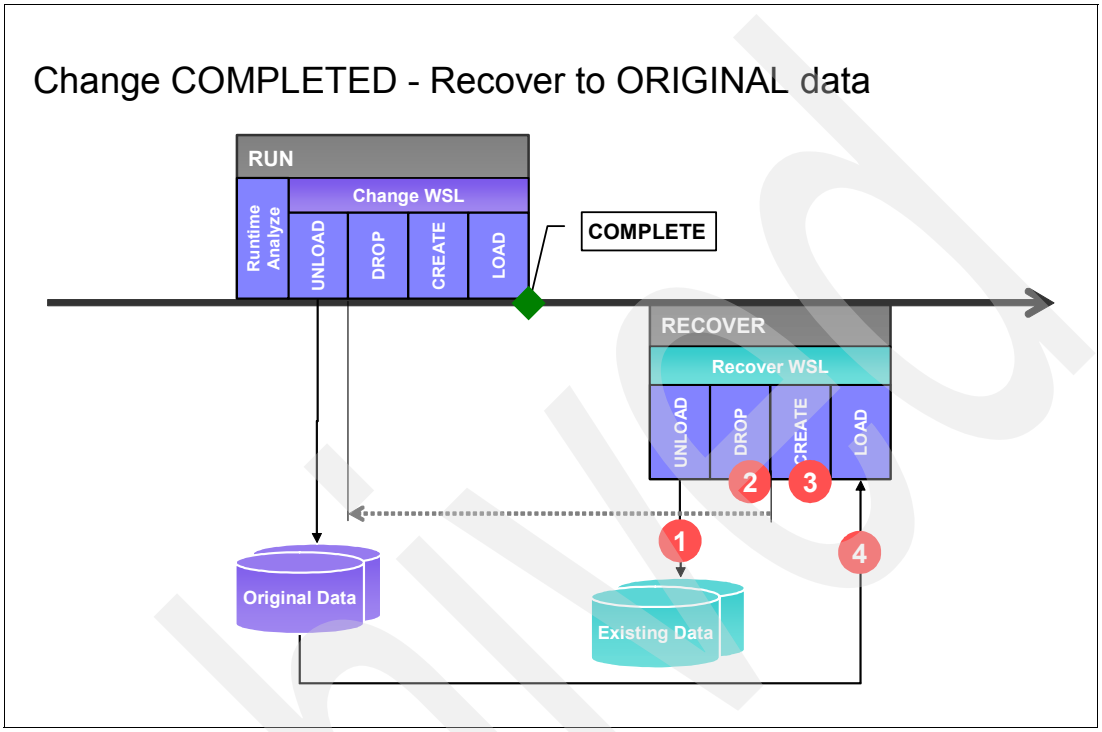

Figure 8-28 Recover a completed change with original data

Everything else stays the same (same table, same structure, same change). This makes it easier to better illustrate the differences in these recovery scenarios.

We start with the same table and define the same changes with ALC that we did the last time. This results in the REGISTER process (of change CH92-01), and first we take a look at the statements that were registered by ALC. We do this to confirm that we have the changes we require and then proceed further. See Figure 8-29.

```
DB2 Admin ----------------- CM - Changes ------------------------ Row 1 of 88
Command ===>                                                  Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel        ID Owner    Name                Type    Status   Comment
            * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
ST       190 PAOLOR2  CH92-01             CHANGE  DEFINED  1ST CHANGE TO TABLE
```

Figure 8-29 Changes panel

So we first issue **ST** against the change CH92-01 and then we get to see all the change statements for that change (see Figure 8-30).

```
DB2 Admin ------------- CM - Change Statements -------------------- Row 1 of 4
Command ===>                                                   Scroll ===> CSR

Change statements for change "PAOLOR2"."CH92-01"
Line commands:
 E - Edit  D - Delete  I - Insert  S - Show

Sel    Sequence O  Qual     Name                Statement
              * *  *        *                   *
--- ----------- -- -------- ------------------ ------------------------------>

              1 TB DSG24RXX GLWTSFN             ALTER TABLE "DSG24RXX"."GLWTSFN
              2 TB DSG24RXX GLWTSFN             ADMIN ALTER TABLE "DSG24RXX"."G
              3 VW DSG24RXX WLSEXFIRSTNAME      ADMIN DROP VIEW  "DSG24RXX"."WL
              4 VW DSG24RXX WLSEXFIRSTNAME       CREATE VIEW DSG24RXX.WLSEXFIRS
****************************** END OF DB2 DATA ******************************
```

Figure 8-30 Changes Statements panel showing all registered changes

This tells us we have all the changes defined that we expect and then we can continue to ANALYZE the change. So we PF3 out of this panel and then issue **AN** against the change. This brings us to the Generate Analyze Job panel. There, for **Generate a recover change,** we select **YES** and then we specify to recover with original data by selecting option **O** (see Figure 8-31).

```
DB2 Admin ----------------- Generate Analyze Job  ----------------------- 19:46
Command ===>

 Specify the following for Analyze:
                                                                  More:     +
 Base version method . . . . . A        (Auto, User, or Existing)

 Change reporting options  . . NO       (Yes/No)

 Required data set information:
   PDS for WSL . . . . . . . 'ADMIN.CM.WSL'
   PDS for batch jobs  . . . 'ADMIN.CM.JCL'
   Prefix for data sets  . . PAOLOR2

 Options:
   Run SQLID . . . . . . . . <NONE>   (Blank, a SQLID, or <NONE>)
   Validate WSL  . . . . . . YES      (Yes/No)
   Use utility options . . . NO       (Yes/No)
   Generate templates  . . . NO       (Yes/No)
   Build JCL to run WSL  . . NO       (Yes/No)
   Generate a recover change YES      (Yes/No)
     Data to recover . . . . O        (Original or Existing)
```

Figure 8-31 Generate Analyze Job panel with recover to original data

Because we have selected to generate a *recover change*, we then get prompted by the register options. This is because recover requires a change ID of its own and hence we have to specify the name and comment (Figure 8-32).

```
DB2 Admin -------------------- CM - Register Options -------------------- 19:40
Option ===>

Commands: CONTINUE                                        DB2 System: DB8A
                                                          DB2 SQL ID: PAOLOR2
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR2  >  (Optional, default is PAOLOR2)
Name  . . . . . . . . RC92-01                                            >
Comment . . . . . . . RECOVER 1ST CHANGE FOR DSG24RXX                    >
Change Type . . . . : RECOVER  (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                      Owner         Name
Ignore  . . . . . . .         >                        >  (? to look up)
Mask  . . . . . . . .         >                        >  (? to look up)
Delta Version . . . . PAOLOR2  >    RC92-01             >  (? to look up)
```

Figure 8-32 Register Options panel for recover change

We keep the naming for the recover change (RC92-01) aligned to the original change (CH92-01). After this we get the analysis job generated and we submit it. This produces the work statement list. C0000190 contains the change that we want to implement.

We look at the work statement list to see how the unload data set appears. In our case it is **PAOLOR2.DB8A.C0000190.ULD.T000.** This data set is then used later in the recover WSL because we specified to recover with original data. See Figure 8-33.

```
DB2 Admin ----------- Show Work Statement List: C0000190 --------- Row 1 of 63
Command ===>                                                  Scroll ===> CSR

Line commands:
 D - Delete  I - Insert  E - Edit  C - Copy  M - Move  A - After  B - Before
 R - Repeat

Select Type Statement
       *    *
------ ---- ------------------------------------------------------------------>
       COM  -- EDITED BY PAOLOR2  ON 2007/03/13 AT 20:01
       COM  -- Created by PAOLOR2 on 2007/03/13 at 19:55
       COM  -- Generated during change management analyze for change ID = 190
       COM  -- Generated by Compare Apply by PAOLOR2 on 2007/03/13 at 19:55
       ADM  PARALLEL UNLOAD
       ADM  JOB
       DB2  -STA DB(DSG24RXX) SPACE(GLWS001) ACCESS(RO)
       ADM  TSODELETE 'PAOLOR2.DB8A.C0000190.CNT.T0001'
       ADM  TSODELETE 'PAOLOR2.DB8A.C0000190.ULD.T0001'
       UTL  TEMPLATE UTLPUNCH DSN 'PAOLOR2.DB8A.C0000190.CNT.T0001'..         U
       UTL  TEMPLATE UTLREC DSN 'PAOLOR2.DB8A.C0000190.ULD.T0001'..         UNI
       UTL  UNLOAD DATA FROM TABLE.."DSG24RXX"."GLWTSFN"..UNLDDN UTLREC..PUNCHD
```

Figure 8-33 Work statement list showing the unload data set name

We then go into the work statement list for the recover change (R0000190) and we want to verify how this work statement list uses the data set that contains the original data (**'PAOLOR2.DB8A.C0000190.ULD.T0001'**). In the RELOAD section of the work statement list we then find this information (Figure 8-34).

```
DB2 Admin ----------- Show Work Statement List: R0000190 -------- Row 40 of 67
Command ===>                                                  Scroll ===> CSR

Line commands:
 D - Delete  I - Insert  E - Edit  C - Copy  M - Move  A - After  B - Before
 R - Repeat

Select Type Statement
       *    *
------ ---- ------------------------------------------------------------------>
       UTL  TEMPLATE SYSREC DSN 'PAOLOR2.DB8A.R0000190.ULD.T0001'..         DIS
       UTL  TEMPLATE UTLDISC DSN 'PAOLOR2.DB8A.R0000190.SDISC.T0001'..
       UTL  TEMPLATE UTLERR DSN 'PAOLOR2.DB8A.R0000190.SERR.T0001'..         UN
       UTL  TEMPLATE UTLMAP DSN 'PAOLOR2.DB8A.R0000190.SMAP.T0001'..         UN
       UTL  TEMPLATE UTLUT1 DSN 'PAOLOR2.DB8A.R0000190.SUT1.T0001'..         UN
       UTL  TEMPLATE UTLOUT DSN 'PAOLOR2.DB8A.R0000190.SOUT.T0001'..         UN
       UTL  UTILFROM PAOLOR2.DB8A.C0000190.CNT.T0001..  ADD(..  SORTNUM 8..  SO
       ADM  ENDJOB
       ADM  ENDPARALLEL
       ADM  CHECKBEGIN
```

Figure 8-34 Work statement list for the recover change using the original data set

We then Edit the highlighted line to see what its contents are (see Figure 8-35).

```
000001 UTILFROM PAOLOR2.DB8A.C0000190.CNT.T0001
000002   ADD(
000003   SORTNUM 8
000004   SORTDEVT SYSDA
000005   WORKDDN(UTLUT1,UTLOUT) ERRDDN(UTLERR)
000006   DISCARDDN(UTLDISC) MAPDDN(UTLMAP)
000007   )
000008   ORIGCNTL(PAOLOR2.DB8A.C0000190.CNT.T0001)
```

Figure 8-35 Recover Work statement list content for UTL UTILFROM line

At this time this data set does not yet exist — it is created during the change when the unload is executed. Then the file contains the load control statements and also the name of the original unload data set name.

Attention: All these files are created dynamically on execution. This means that you cannot see the generated LOAD control statements before the change has happened.

So we run the change and then check whether it has completed. When the change has completed we then can look into the data set **PAOLOR2.DB8A.C0000190.CNT.T0001** (see Figure 8-36).

```
 BROWSE    PAOLOR2.DB8A.C0000190.CNT.T0001          Line 00000000 Col 001 080
 Command ===>                                              Scroll ===> CSR
********************************** Top of Data ********************************
TEMPLATE BB4ACJI8
     DSN('PAOLOR2.DB8A.C0000190.ULD.T0001')
     DISP(OLD,KEEP,KEEP)
LOAD DATA INDDN BB4ACJI8 LOG NO  RESUME YES
 EBCDIC  CCSID(00037,00000,00000)
 SORTKEYS                 84
 INTO TABLE "DSG24RXX"."GLWTSFN"
 WHEN(00001:00002) = X'003C'
 ( "SFN_NO"
        POSITION(  00003:00004) SMALLINT
 , "SEX"
        POSITION(  00005:00005) CHAR(00001)
 , "FIRSTNAME"
        POSITION(  00006:00019) VARCHAR
 )
******************************** Bottom of Data ******************************
```

Figure 8-36 Data set generated by change and to be used from recover change

This data set **PAOLOR2.DB8A.C0000190.CNT.T0001** is used in the recover work statement list. And we can see that it contains the actual name of the unload data set which represents the old data **PAOLOR2.DB8A.C0000190.ULD.T0001.** Then we can go back to the changes panel (see Figure 8-37).

```
DB2 Admin ------------------ CM - Changes ------------------------ Row 1 of 89
Command ===>                                                  Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
         191 PAOLOR2  RC92-01             RECOVER ANALYZED RECOVER 1ST CHANGE FO
RC       190 PAOLOR2  CH92-01             CHANGE  COMPLETE 1ST CHANGE TO TABLE
```

Figure 8-37 Changes panel showing that the CHANGE is COMPLETE

We issue **RC** against the CHANGE (not the RECOVER) to recover the change. This brings us then to an edit session with the generated recover JCL (see Figure 8-38).

```
DB2 Admin -------------- Edit Generated JCL -------------- Columns 00001 00072
Command ===>                                                  Scroll ===> CSR

****** ***************************** Top of Data ******************************
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001 //PAOLOR2A JOB (999,POK),'G.PFANN',NOTIFY=&SYSUID,
000002 //*       RESTART=STEPNAME, <== FOR RESTART REMOVE * AND ENTER STEP NAME
000003 //          CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),REGION=0M
000004 /*JOBPARM SYSAFF=SC63
000005 //*
000006 //*
000007 //*
000008 //*********************************************************************
000009 //* DB2 ADMINISTRATION TOOL GENERATED JOB
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
 The recover job can be submitted now.  The recover strategy has determined e
 that there are no completed changes that have to be completed first, and   e
 there are no pending changes that will be set to DEFINED status.  However, e
 the recover strategy is re-calculated at runtime and thus may be different e
 from what it is now.                                                       e
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

Figure 8-38 Generated recovery JCL

There are no pending changes detected this time — remember that in the other scenario with “existing data” we had pending changes and had to change the recover JCL so that ADBTEP2 PARM PCACT got changed — but not this time.

And also, the message says: **there are no completed changes that have to be completed first**.

If we scroll down in the recover JCL, we actually see that the recover work statement list has been copied into the job (see Figure 8-39).

```
****** ***************************** Top of Data ******************************
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  2 Line(s) not Displayed
000003 -- This is a recover change for change ID = 190
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 33 Line(s) not Displayed
000037   CREATE TABLE DSG24RXX.GLWTSFN
000038      (SFN_NO               SMALLINT NOT NULL GENERATED BY DEFAULT
000039         AS IDENTITY
000040           (START WITH 1, INCREMENT BY 1, CACHE 20, NO CYCLE,
000041            NO ORDER, MAXVALUE 32767, MINVALUE -32768),
000042       SEX                  CHAR(1) FOR SBCS DATA NOT NULL
000043         WITH DEFAULT,
000044       FIRSTNAME            VARCHAR(12) FOR SBCS DATA NOT NULL
000045         WITH DEFAULT)
000046     IN DSG24RXX.GLWS001
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 48 Line(s) not Displayed
000095 UTILFROM PAOLOR2.DB8A.C0000190.CNT.T0001
000096   ADD(
000097   SORTNUM 8
000098   SORTDEVT SYSDA
000099   WORKDDN(UTLUT1,UTLOUT) ERRDDN(UTLERR)
000100   DISCARDDN(UTLDISC) MAPDDN(UTLMAP)
000101   )
000102   ORIGCNTL(PAOLOR2.DB8A.C0000190.CNT.T0001);
```

Figure 8-39 Recover JCL with recover work statement list inside the job

This fact is important in case the recover work statement list fails and we have to restart and manually change the recover work statement list.

Tip: Restart of the recover JCL is done with **ER** against the RECOVER change.

Attention: ER does not copy the current work statement list from **ADMIN.CM.WSL**. In this case we have to change the WSL inside the recover JCL (and not in the work statement list data set **ADMIN.CM.WSL**).

So we are ready to go and submit the JCL. After the recover change has finished, we can see the result on the CM changes panel (Figure 8-40).

```
DB2 Admin ------------------ CM - Changes ------------------------ Row 1 of 89
Command ===>                                                   Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
         191 PAOLOR2  RC92-01            RECOVER COMPLETE RECOVER 1ST CHANGE FO
         190 PAOLOR2  CH92-01            CHANGE  DEFINED  1ST CHANGE TO TABLE
```

Figure 8-40 CM Changes panel showing the recover in COMPLETE status

So when we have reached this point, we are back on V0 (with our database structure) and we are back with the old data. Mission completed!

8.3 Recovering an uncompleted change

In the following scenario we describe the actions you have to take to be able to recover an uncompleted change. The Change WSL execution fails, and we have to check the job output to determine the cause of failure and make the necessary corrections before restart the modified job. If the job fails anywhere before the CREATE step, you can restart the job at the beginning. In our scenario, the job failed after the DROP step, therefore modifications in the JCL are necessary (see Figure 8-41).

The following actions are necessary:

1. Modify abnormally terminated Recover WSL
2. Re-run modified Recover WSL

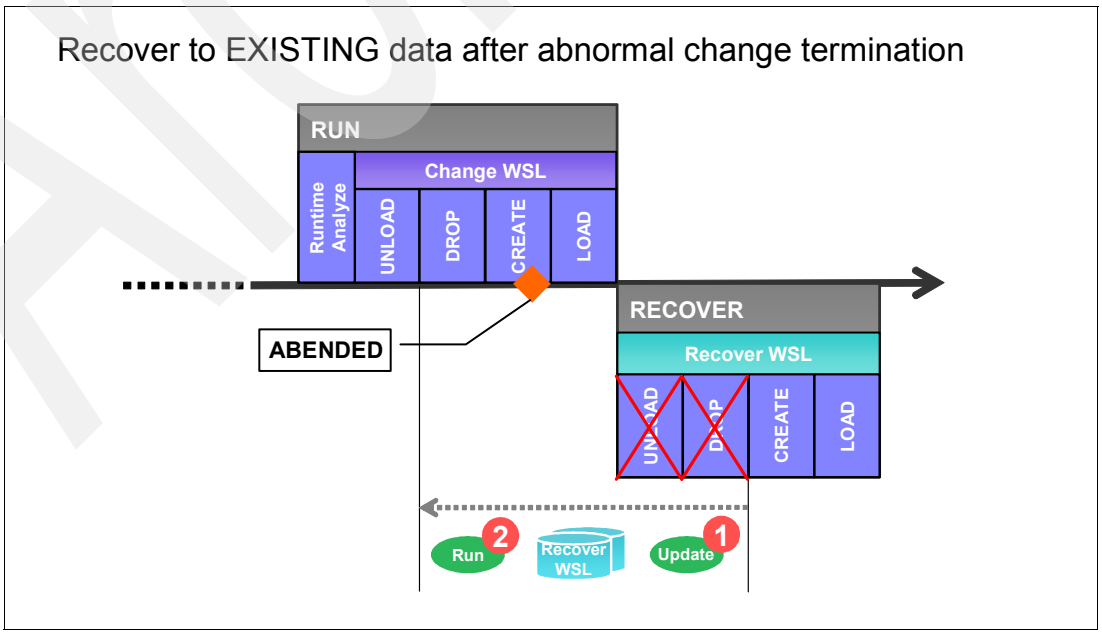

Figure 8-41 Recover to existing data after change abnormally terminated

In the Changes panel, as shown in Figure 8-42, two changes are displayed. The recover change (ID 229) was registered during the analyze process, and the Recover WSL was successfully generated. The change is placed in ANALYZED status.

The original change (ID 228), which is in RUNNING status, does not change to COMPLETE status, which indicates that the job failed at some point. We cannot issue the **RC** line command to recover the change because it is not in COMPLETE status.

```
DB2 Admin ------------------ CM - Changes ------------------- Row 209 from 210
 Command ===>                                                  Scroll ===> CSR

 Line commands:
  U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
  PQ - Prerequisites  IG - Ignores  MA - Masks
  ? - Show all line commands

 Sel       ID Owner    Name                Type    Status   Comment
            * *        *                   *       *        *
 --- -------- -------- ------------------- ------- -------- -------------------->
          228 PAOLOR3  D0742007 ORIG ALC  CHANGE  RUNNING  ORIGINAL CHANGE FOR C
          229 PAOLOR3  D0742007 RECV ALC  RECOVER ANALYZED RECOVER CHANGE FOR CH
****************************** END OF DB2 DATA ******************************
```

Figure 8-42 Changes panel - Original change in status RUNNING and recover change in status ANALYZED

Fixing the problem

In the following several figures, we take you through our troubleshooting process.

As we can see in the job output in Figure 8-43 on page 356, the root cause for the abnormally termination of the original change was a create of an index on a non-existent table.

As the next step, we have to modify the recover job as shown in Figure 8-44 on page 357. Therefore we edit the CM recover JCL on the data set ADMIN.CM.JCL(R0000228) and make the following updates:

- We set the UNLOAD related rows in comment, as we have no current data to load and we have to recover to existing data.
- We put the DROP statement in comment.
- To confirm the supersede action, we set the PCACT parameter from CANCEL (default) to SUPERSEDE. Otherwise the recover job is not run.

After these JCL modifications have been done, we submit the job, which is listed in Figure 8-44 on page 357 and Figure 8-45 on page 358.

```
000001                    J E S 2  J O B  L O G  --  S Y S T E M  S C 6 3  --
000002
000003  16.52.16 JOB21851 ---- THURSDAY,  15 MAR 2007 ----
000004  16.52.16 JOB21851  IRR010I  USERID PAOLOR3  IS ASSIGNED TO THIS JOB.
000005  16.52.16 JOB21851  IEF677I WARNING MESSAGE(S) FOR JOB PAOLOR3A ISSUED
000006  16.52.16 JOB21851  ICH70001I PAOLOR3  LAST ACCESS AT 16:45:11 ON THURSD
000007  16.52.16 JOB21851  $HASP373 PAOLOR3A STARTED - INIT 1    - CLASS A - SY
000008  16.52.16 JOB21851  IEF403I PAOLOR3A - STARTED - TIME=16.52.16 - ASID=00
000009  16.52.17 JOB21851  -                                          --TIMINGS
000010  16.52.17 JOB21851  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    S
000011  16.52.17 JOB21851  -PAOLOR3A          EXDLTA      00    337    .00    .
000012  16.52.17 JOB21851  -PAOLOR3A          EXOBJS1     00    160    .00    .
000013  16.52.20 JOB21851  -PAOLOR3A          GNSQL2      00    477    .00    .
000014  16.52.22 JOB21851  -PAOLOR3A          MERGE3      00   1213    .00    .
000015  16.52.23 JOB21851  -PAOLOR3A          COMP4       00    579    .00    .
000016  16.52.25 JOB21851  -PAOLOR3A          GAPY5       00    653    .00    .
000017  16.52.26 JOB21851  -PAOLOR3A          WSLCM6      00    669    .00    .
000018  16.52.27 JOB21851  -PAOLOR3A          COMP7       00    571    .00    .
000019  16.52.28 JOB21851  -PAOLOR3A          GAPY8       00    653    .00    .
000020  16.52.29 JOB21851  -PAOLOR3A          WSLCM9      00    672    .00    .
000021  16.52.29 JOB21851  -PAOLOR3A          DB2B10      08    181    .00    .
000022  16.52.30 JOB21851  IEF404I PAOLOR3A - ENDED - TIME=16.52.30 - ASID=0031
000023  16.52.30 JOB21851  -PAOLOR3A ENDED.                          TOTAL CP
000024  16.52.30 JOB21851  $HASP395 PAOLOR3A ENDED
000025 0------ JES2 JOB STATISTICS ------
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -   940 Line(s) not Displayed
000966  ADB5044E *** INPUT STATEMENT:
000967     CREATE UNIQUE INDEX DSG24RXX.GLWXSFN1
000968       ON DSG24RXX.GLWTSFN
000969        (SFN_NO               ASC)
000970       USING STOGROUP GLWG01
000971       PRIQTY 12 SECQTY 12
000972       ERASE  NO
000973       FREEPAGE 0 PCTFREE 10
000974       GBPCACHE CHANGED
000975       CLUSTER
000976       BUFFERPOOL BP16
000977       CLOSE YES
000978       COPY NO
000979       DEFINE YES
000980       PIECESIZE 2 G;
000981  sqlerror on CREATE     command, PREPARE    function
000982   DSNT408I SQLCODE = -204, ERROR:  DSG24RXX.GLWTSFN IS AN UNDEFINED NAME
000983   DSNT418I SQLSTATE   = 42704 SQLSTATE RETURN CODE
000984   DSNT415I SQLERRP    = DSNXODDL SQL PROCEDURE DETECTING ERROR
000985   DSNT416I SQLERRD    = 35  0  0  -1  0  0 SQL DIAGNOSTIC INFORMATION
000986   DSNT416I SQLERRD    = X'00000023'  X'00000000'  X'00000000'  X'FFFFFFF
000987            X'00000000'  X'00000000' SQL DIAGNOSTIC INFORMATION
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -   450 Line(s) not Displayed
****** **************************** Bottom of Data ****************************
```

Figure 8-43 Failed CM run job for change Id 228

```
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
EDIT       ADMIN.CM.JCL(R0000228) - 01.03                 Columns 00001 00072
Command ===>                                                 Scroll ===> CSR
****** ***************************** Top of Data ******************************
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 33 Line(s) not Displayed
000034 //DB2B1 EXEC PGM=IKJEFT01,DYNAMNBR=100
000035 //STEPLIB  DD DISP=SHR,DSN=DB8A8.SDSNEXIT
000036 //         DD DISP=SHR,DSN=DB8A8.SDSNLOAD
000037 //SYSEXEC  DD DISP=SHR,DSN=GOC.V7R2M0.SGOCEXEC
000038 //         DD DISP=SHR,DSN=ADB.V7R2M0.SADBEXEC
000039 //MSGLIB   DD DISP=SHR,DSN=GOC.V7R2M0.SGOCLLIB
000040 //         DD DISP=SHR,DSN=ADB.V7R2M0.SADBLLIB
000041 //SYSTSPRT DD SYSOUT=*
000042 //SYSPRINT DD SYSOUT=*
000043 //UTPRINT  DD SYSOUT=*
000044 //RNPRIN01 DD SYSOUT=*
000045 //ADBDIAG  DD SYSOUT=*
000046 //SYSTSIN  DD *
000047  DSN SYSTEM(DB8A)
000048  RUN  PROGRAM(ADBTEP2) PLAN(A72) -
000049    LIB('ADB.V7R2M0.SADBLLIB')  -
000050    PARMS('/WORKLIST(R0000228) SSID(DB8A)  -
000051  RESTART(YES),RECOVER(228),PCACT(SUPERSEDE)')
000052  END
000053 //SYSIN    DD *
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  4 Line(s) not Displayed 000054
-- Created by PAOLOR3 on 2007/03/15 at 16:45
000058 -STA DB(DSG24RXX) SPACE(GLWS001) ACCESS(RO);
000059 --
000060 -- TSODELETE 'PAOLOR3.DB8A.R0000228.CNT.T0001';
000061 -- TSODELETE 'PAOLOR3.DB8A.R0000228.ULD.T0001';
000062 -- TEMPLATE UTLPUNCH DSN 'PAOLOR3.DB8A.R0000228.CNT.T0001'
000063 --          UNIT SYSDA;
000064 -- TEMPLATE UTLREC DSN 'PAOLOR3.DB8A.R0000228.ULD.T0001'
000065 --          UNIT SYSDA;
000066 -- UNLOAD DATA FROM TABLE
000067 -- "DSG24RXX"."GLWTSFN"
000068 -- UNLDDN UTLREC
000069 -- PUNCHDDN(UTLPUNCH);
000070 -- TSODELETE 'PAOLOR3.DB8A.R0000228.CNC.T0001';
000071 -- ALLOC DD(INTFIN)
000072 --       DS('PAOLOR3.DB8A.R0000228.AN164505.IFF(IFF0001)') SHR;
000073 -- ALLOC DD(DDLIN) DUMMY;
000074 -- ALLOC DD(DDLOUT) DUMMY;
000075 -- ALLOC DD(C001)
000076 --       DS('PAOLOR3.DB8A.R0000228.CNT.T0001') SHR;
000077 -- ALLOC DD(D001)
000078 --       DS('PAOLOR3.DB8A.R0000228.CNC.T0001')
000079 --       LIKE('PAOLOR3.DB8A.R0000228.CNT.T0001');
000080 -- ADBSYSIN COPYDD(ADBIN001);
000081 -- RUN PROGRAM(ADB27AC) PLAN(A72)
000082 --      LIB('ADB.V7R2M0.SADBLLIB');
000083 --   DROP TABLE DSG24RXX.GLWTSFN ;
000084 --   COMMIT ;
000085 -STA DB(DSG24RXX) SPACE(GLWS001);
000086   CREATE TABLE DSG24RXX.GLWTSFN
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 35 Line(s) not Displayed
```

Figure 8-44 Change Management recover job (1/2)

Archived

```
000122 TSODELETE 'PAOLOR3.DB8A.R0000228.SDISC.T0001';
000123 TSODELETE 'PAOLOR3.DB8A.R0000228.SERR.T0001';
000124 TSODELETE 'PAOLOR3.DB8A.R0000228.SMAP.T0001';
000125 TSODELETE 'PAOLOR3.DB8A.R0000228.SUT1.T0001';
000126 TSODELETE 'PAOLOR3.DB8A.R0000228.SOUT.T0001';
000127 -- TEMPLATE SYSREC USED IF HPU UNLOAD
000128 -- TEMPLATE SYSREC DSN 'PAOLOR3.DB8A.R0000228.ULD.T0001'
000129 --         DISP(SHR,KEEP,KEEP);
000130 TEMPLATE UTLDISC DSN 'PAOLOR3.DB8A.R0000228.SDISC.T0001'
000131          UNIT SYSDA
000132          SPACE(1,50) CYL;
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 51 Line(s) not Displayed
000184 -- End of Compare Apply statements
000185 /*
000186 //ADBIN001 DD *
000187 TC='DSG24RXX',TN='GLWTSFN',TCN='DSG24RXX',TNN='GLWTSFN';
****** **************************** Bottom of Data ****************************
```

Figure 8-45 Change Management recover job (2/2)

After submitting the job, the batch restart program ADBTEP2 finds a record in the checkpoint table which indicates that a restart is required.

The CREATE statements are executed and the existing data is reloaded. As we have specified LOAD LOG NO and the affected objects are now COPY PENDING status, the job ends with a RC 04.

After successful execution of the job, the recover change is placed in COMPLETE status as shown in Figure 8-46.

```
DB2 Admin ------------------ CM - Changes ------------------- Row 111 from 114
Command ===>                                                  Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
           * PAOLOR3* *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
CAN      228 PAOLOR3  D0742007 ORIG ALC  CHANGE  RUNNING  ORIGINAL CHANGE FOR C
         229 PAOLOR3  D0742007 RECV ALC  RECOVER COMPLETE RECOVER CHANGE FOR CH
****************************** END OF DB2 DATA ******************************
```

Figure 8-46 Changes panel - Recover change in status COMPLETE

The original change is still set to RUNNING. We want to cancel this change using the **CAN** line command. In consequence of this action, the status of the recover change is also set to CANCELED (see Figure 8-47).

```
DB2 Admin ------------------ CM - Changes ------------------- Row 209 from 212
Command ===>                                                  Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                 Type    Status   Comment
           * PAOLOR3* *                    *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
         228 PAOLOR3  D0742007 ORIG ALC  CHANGE  CANCELED ORIGINAL CHANGE FOR C
         229 PAOLOR3  D0742007 RECV ALC  RECOVER CANCELED RECOVER CHANGE FOR CH
******************************** END OF DB2 DATA ********************************
```

Figure 8-47 Changes panel - Original and recover change in status CANCELED

8.4 Backing out to a previous version (without RECOVER process)

In this scenario we show you how to back out to a base version (BVn) which was made in an earlier phase. This is always done with existing data (see Figure 8-48).

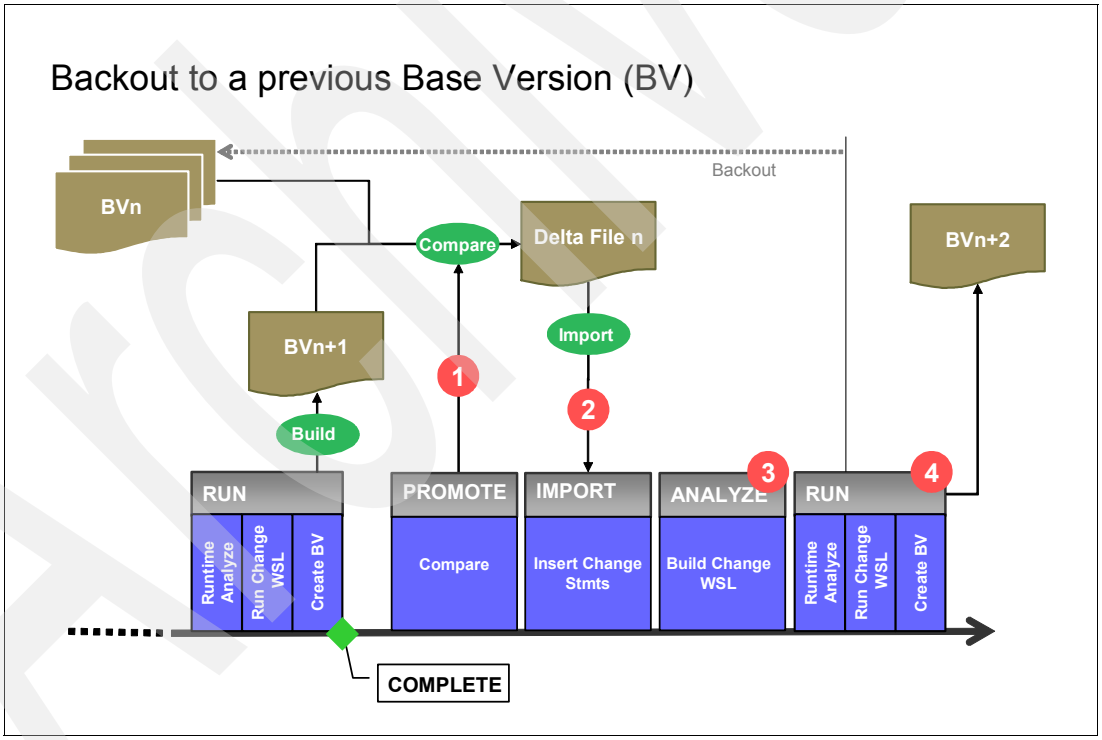

Figure 8-48 Backout to a previous base version

The following actions are necessary:

1. Compare base version and create a delta version file
2. Import delta change statements
3. Analyze promote change and generate Change WSL
4. Execute Change WSL

On the Promote panel, as shown in Figure 8-49, we specify the two base versions (BVn and BVn+1) that we want to compare to produce a delta file data set (DFn) that contains the statements that represent the differences.

We enter owner and name of the starting version (BVn). This version contains a snapshot of the objects before changes were made. In the ending version fields we specify the BVn+1 which contains a snapshot of the objects after the changes we wish to promote. The Delta changes data set identifies where the delta change statements are saved.

.

```
DB2 Admin ----------------------- CM - Promote -------------------------- 15:14
Command ===>

Starting Version (Old):
  Owner . . . . . . PAOLOR3  >            (? to look up)
  Name  . . . . . . BVN                >  (? to look up)

 or enter a data set name that contains a Start Version:

  Data set name . .

Ending Version (New):
  Owner . . . . . . PAOLOR3  >            (? to look up)
  Name  . . . . . . BVN+1              >  (? to look up)

Output data set names:
  Promote JOB JCL . ADMIN.CNTL
  Delta change  . . ADMIN.CM.DVFN
```

Figure 8-49 Promote panel - create delta file

We press Enter, which brings us to the Register Options panel listed in Figure 8-50. We specify owner and name of the change and in the comment filed optional remarks. We type the primary command CONTINUE to go on.

```
DB2 Admin -------------------- CM - Register Options -------------------- 15:16
Option ===> CONTINUE

Commands: CONTINUE                                   DB2 System: DB8A
                                                     DB2 SQL ID: PAOLOR3
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR3  >  (Optional, default is PAOLOR3)
Name  . . . . . . . . D0742007 COMPARE BASE VERSIONS                      >
Comment . . . . . . . COMPARE BVN WITH BVN+1                              >
Change Type . . . . : COMPARE  (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                     Owner         Name
Ignore  . . . . . . .         >                           >  (? to look up)
Mask  . . . . . . . .         >                           >  (? to look up)
Delta Version . . . .         >                           >  (? to look up)
```

Figure 8-50 Register options panel for compare change

JCL is generated that contains compare steps and a register step for the compare change. We submit the batch job, the delta file is created, and the change is registered in the Change Management database (see Figure 8-51).

```
DB2 Admin ------------------ CM - Changes ------------------- Row 173 from 173
Command ===>                                                  Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
         218 *        *                   *       *        *
--- -------- -------- -----------------> ------- -------- -------------------->
         218 PAOLOR3  D0742007 COMPARE B COMPARE COMPLETE COMPARE BVFN WITH BVF
****************************** END OF DB2 DATA ******************************
```

Figure 8-51 Changes panel - Compare change in CAN

The delta changes data are now imported as a new change as shown in Figure 8-52. The Import Changes panel allows us to import the SQL statements into a new change. We specify the name of the input data set that was generated by promoting a change with DB2 Admin.

```
DB2 Admin ------------------- CM - Import Changes ---------------------- 15:33
Command ===>


Input data set information:                          DB2 System: DB8A
 Data set name 'ADMIN.CM.DVFN'

Prereq resolution mode  . TSO       (TSO/Batch)
Execution mode  . . . . . TSO       (TSO/Batch)
```

Figure 8-52 Import Changes panel - Specify input data set

On the Register Options panel, as shown in Figure 8-53, we are prompted to specify owner, name, and comment for the PROMOTE change.

```
DB2 Admin -------------------- CM - Register Options -------------------- 15:38
Option ===> CONTINUE

Commands: CONTINUE                                   DB2 System: DB8A
                                                     DB2 SQL ID: PAOLOR3
Specify the following values to register a change:

Owner . . . . . . . . PAOLOR3  >  (Optional, default is PAOLOR3)
Name  . . . . . . . . D0742007 DVFN                                   >
Comment . . . . . . . PROMOTE DELTA VERSION FILE N                    >
Change Type . . . . . : PROMOTE  (Promote, Change, Compare, Recover, Fast)

Specify the owner and name values to use for this change:
                      Owner         Name
Ignore  . . . . . . .         >                           >  (? to look up)
Mask  . . . . . . . .         >                           >  (? to look up)
Delta Version . . . . PAOLOR3  >    D0742007 DVFN         >  (? to look up)
```

Figure 8-53 Register Options panel for promote change

We now enter CONTINUE as primary command and the promote change has been created and registered in the CM database. See Figure 8-54.

```
DB2 Admin ------------------ CM - Changes ------------------- Row 208 from 208
Command ===>                                                   Scroll ===> CSR

Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks
 ? - Show all line commands

Sel       ID Owner    Name                Type    Status   Comment
         227 *        *                   *       *        *
--- -------- -------- ------------------ ------- -------- -------------------->
         227 PAOLOR3  D0742007 DVFN      CHANGE  DEFINED  PROMOTE DELTA VERSION
****************************** END OF DB2 DATA ******************************
```

Figure 8-54 Changes panel - promote change in status defined

The change is now in DEFINED status and is ready to be analyzed. We go through the ANALYZE process to validate the change and to generate a WSL.

In the RUN process we have the option to create a new base version (BVn+2) of the objects after the change have been made successfully.

We can now apply the changes by submitting the generated batch job.

After successful execution, the change is placed in COMPLETE status.

8.5 Drop Recovery

Often there is a requirement to retain the DDL of objects that get dropped during a WSL execution. This comes into play when, for whatever reason, the WSL fails during the execution. Such an event sometimes leaves the customer with a decision to re-create the objects as they were before the WSL started.

One thing that you can always do is to work with RECOVER YES when you ANALYZE the change. This gives you a WSL which contains the reverse action of your change. An example is shown Figure 8-55.

```
ALTER TABLE "DSG24RJS"."GLWTVRN"   ADD "NEWAGAINNN"   SMALLINT   NOT NU
LL WITH DEFAULT
```

Figure 8-55 Change statement taken from the change statement table

The resulting WSL statement is shown in Figure 8-56.

```
....
DDL    ALTER TABLE DSG24RJS.GLWTVRN
DML    COMMIT
....
```

Figure 8-56 Extract of work statement list content showing the DDL part

The Recover change statement is shown in Figure 8-57.

```
ADMIN ALTER TABLE DSG24RJS.GLWTVRN DROP COLUMN NEWAGAINNN
```

Figure 8-57 Change statement taken from the change statement table for the recover change

The Recover change WSL is shown in Figure 8-58.

```
....
DDL    CREATE TABLE DSG24RJS.GLWTVRN
DDL    COMMENT ON TABLE DSG24RJS.GLWTVRN
DML    COMMIT
DDL    CREATE UNIQUE INDEX DSG24RJS.GLWXVRN1
DML    COMMIT
DDL    SET CURRENT PATH = "SYSIBM","SYSFUN","SYSPROC","DB2ADM"
DDL    CREATE VIEW DSG24RJS.WLVERSN AS
....
```

Figure 8-58 Extract of work statement list content showing the DDL part

In fact, we have provided a scenario (see Figure 8-41 on page 354) which shows in essence such a drop recovery, and it also contains information to bring the old data back to the old structure.

The DDL is stored in the work statement list among all other activities (unload, drop, load, and so on). If you want to work on the DDL alone, you have to pull it out of the work statement list into a separate file.

But be aware — this DDL inside the WSL is generated at ANALYZE time. So any change to the object structure later outside the CM control goes unnoticed; it can mean that the DDL might not be valid anymore.

Important: The DDL is not generated dynamically when the DROP occurs.

Archived

Archived

9

Additional considerations

In this chapter we provide information about various topics for both Admin and CM. We cover general issues (such as troubleshooting) that deal with Object Compare, Admin, and CM, as well as hints and tips about lesser known functions of Change Management.

We cover the following topics:

- Performance considerations (scalability)
- Diagnosis and troubleshooting
- Data conversion
- Limitations with CM

Archived

© Copyright IBM Corp. 2007. All rights reserved.

9.1 Performance considerations (scalability)

For improved performance, you should do frequent RUNSTATS on both the DB2 table spaces and Change Management table spaces. This should be done on a regular basis depending on the amount of changes to the catalog. With large amounts of changes, you should do more frequent RUNSTATS. The Change Management database is ADBDCHG and the nine (9) table spaces are as follows:

- ADBSCHG
- ADBSCHGS
- ADBSCPRQ
- ADBSCMSK
- ADBSCIGN
- ADBSCVER
- ADBSCVLN
- ADBSCVSC
- ADBSCID

The foregoing tablespaces should be added to the customer's normal RUNSTATS, which are done on a regular basis in their shop.

Another important factor for improving performance in the CM process is to limit the number of outstanding or pending changes affecting a new change. These are changes that have been registered but are incomplete or have a status of defined, analyzed, or running. To remove these pending changes, they should be analyzed and run to completion.

9.1.1 Large CM database

If the CM database grows too large and you would like to delete version files, they can be deleted manually or by a user-written SQL statement. You can also use RI cascade delete to clean up dependent tables. We highly recommend that you obtain a backup copy of the DM database before performing any of these processes.

In general, the indexes already part of Change Management prevent performance from being affected by the size of the CM database tables.

9.1.2 Additional indexes

There is a performance issue for some users who run ADB2GEN on DB2 systems that have no aliases defined. The Object Comparison Tool has performance issues when processing large object sets in DB2 V7 and V8 if the cardinality for index DSNDTX03 (columns TBCREATOR and TBNAME) on table SYSIBM.SYSTABLES is low.

You should apply PTF for PK37849. It has the create indexes within an ADBCX sample. These indexes work well even if no aliases and/or synonyms are defined. DB2 Admin and Object Compare test for the existence of aliases and synonyms. If there are none, some queries in ADB2GEN are bypassed.

ADBCX has the following CREATEINDEX (type 2) statements for DB2 V8 and later:

- "SYSIBM"."DBRMX" on "SYSIBM"."SYSDBRM"
- "SYSIBM"."STMTX" on "SYSIBM"."SYSSTMT"
- "SYSIBM"."FIELDSX" on "SYSIBM"."SYSFIELDS"
- SYSIBM"."TABLESXS" on "SYSIBM"."SYSTABLES"

- "SYSIBM"."COLUMNSX" on "SYSIBM"."SYSCOLUMNS"
- "SYSIBM"."TABLESX" on "SYSIBM"."SYSTABLES"
- "SYSIBM"."DEPSX" on "SYSIBM"."SYSVIEWDEP"

These indexes are beneficial for DB2 V7 but are not necessary for DB2 V8 and later:

- "SYSIBM"."FORKEYSX" on "SYSIBM"."SYSFOREIGNKEYS"
- "SYSIBM"."RELSX" on "SYSIBM"."SYSRELS"

9.1.3 Virtual changes

Pending changes are treated as virtual changes when you make changes by the following methods:

- The Alter dialogs:
  - To rename a database (AL RENAME)
  - To redefine a table space (AL RDEF)
  - To redefine an index (RDEF)
  - To redefine a table (ALC)
- The Create dialogs (option 2.4 from the DB2 Admin main menu):
  - To create a table space
  - To create a table
  - To create an index
  - To create a materialized table
  - To create a view
  - To create a trigger

When you are defining a new change, you are asked how to deal with the pending changes. If you indicate that you want to apply the pending changes, then any conflicts or changes to your object are applied as virtual changes. This allows you to define your new changes based on a virtual representation of the objects after the pending changes were performed.

> **Note:** The list of pending changes that DB2 Admin displays might not be complete. However, if you apply the pending changes, all of the pending changes for the objects are applied as virtual changes whether or not they appear in the list.

However, if you request for your new change to supersede the pending changes, you do not see the virtual changes. This new change becomes a prerequisite change for the pending changes (virtual changes).

We recommend that you minimize the number of pending or outstanding changes to improve performance of subsequent changes. To remove pending changes, the change should be analyzed and run until they are complete.

9.1.4 Using Data Archive Expert for z/OS

IBM DB2 Data Archive Expert for z/OS, program number 5697-J22, is a tool that can be used to archive data with Change Management. See *IBM DB2 Data Archive Expert for z/OS: Put Your Data in Its Place*, SG24-SG24-7080-00 and *IBM DB2 Data Archive Expert User's Guide and Reference Version 2 Release 1*, SC18-9696-02.

9.2 Diagnosis and troubleshooting

If you receive a DB2 Admin/OC error message that does not contain adequate information regarding the actions you should take, use the following features to diagnose the problem:

- Abend codes
- Dump - record title and failing module/CSECT name if appropriate
- Printout of the trace back from a Language Environment (LE) dump
- Check to see if maintenance was recently applied that might be affecting your failure
- Check where there any recent changes to the system or application
- VTAM®, MVS, DB2, DB2 Admin or Object Compare Change Management messages

If the above information contains adequate information regarding the actions you should take, see if you can solve the issue.

If this is a Change Management issue, try to simplify the problem. Attempt to recreate using little or no prerequisite or pending changes. If possible, do not use an ignore file.

However, if you cannot determine the solution, contact IBM Support. Gather the following documentation and make it available to the Support Center for your incident report:

- DB2 Admin or Object Compare version number, release number, and maintenance level.
- Which process fails? (Admin, OC, CM, installation, performance, and so on)
- What messages or symptoms are occurring?
- List all files that could be used to recreate the failure:
  - Source DDL and Target DDL
    (specify if obtained from DDL/catalog/automatic target selection)
  - Version files (target, source, or change management)
  - Ignore file
  - Mask file
  - Job output
  - ADBDIAG DEBUG output if available
- Include a dump if available.
- Obtain screen prints of each panel along with the specific options used.
- Determine what objects were a part of the failure.
- Check where there were any prereq or pending changes.
- Complete DDL of the object(s) that failed (obtained from DB2 Admin GEN).

Once documentation has been gathered, it should be sent to the IBM FTP server; see Figure 9-1.

Archived

```
 TESTCASE.BOULDER.IBM.COM
       USER: anonymous
       PASSWORD: your complete e-mail address

Place your file on the s390/toibm/db2 directory using the PUT command for DB2
Tools. When the transfer is complete, provide the following  information in
your ETR:
  File name(s) - Prefix with your PMR number (e.g. 09165.LOG).
  MVS data set attributes including LRECL, RECFM, BLKSIZE, and space.

Terse SVC dumps and PDS's.  Files must be sent RECFM FB.  Log tapes (OLDS) MUST
be tersed before sending via FTP.  Use TRSMAIN to terse a file.  You can obtain
TRSMAIN from: ftp://ftp.software.ibm.com/s390/mvs/tools/packlib/.

NOTE:  If you are getting access denied when trying to FTP files to one of the
toibm directories, verify that the characters ' $ ' and ' ~ ' are not included
in the file name.  Those two characters are not allowed on testcase so the file
will have to be renamed.

For additional information see:
http://www.ibm.com/de/support/ecurep/howto.html or information APAR II11945.
```

Figure 9-1 Sending Doc to FTP, II11945

9.2.1 TSO ISPDPTRC

During some functions of DB2 Admin and Object Compare Change Management, you might be requested to run a Debug trace. If you are on z/OS 1.6 or higher, you can use the command: **TSO ISPDPTRC** to capture a debug file. After entering the **TSO ISPDPTRC** command, you should recreate the failing command or procedure. Then you can re-issue **TSO ISPDPTRC** to stop the trace activity. You can use **v** to browse debug data set and review the trace. Make note of the data set name of the debug trace file. PF3 out of the data set, then FTP this trace data to the Support Center as additional diagnostic data.

This trace shows the processing flow of the panels and skeletons.

For additional information on this trace, see *z/OS V1R7.0 ISPF Dialog Developer's Guide* SC34-4821-04.

9.2.2 ADBDIAG DD statement

DB2 Admin has an internal tool to gather information to help support analyze and diagnose reported problems. Occasionally an Lv2 representative might request that you recreate the failure using this trace facility. The functions that write these diagnostic records normally do it automatically when the ADBDIAG file is allocated and stops when the file reaches end of file (EOF) or fills up. The ADBDIAG file is reused or over-written on each invocation of a component so you should capture and save the contents as soon as possible. Trace data is placed in ADBDIAG a VIO data set. Only the following functions within Admin use this tool:

- Revoke Impact reports — this produces an unusual quantity of trace records
- Space calculation (MIG, ALC, RDEF, OC, UTILs)
- WSL Cloning — written to if an error occurs during cloning
- Change Management (pre-req processing, virtual changes, Import)
- ADBTEP2

ZPARMs - Option z.2z.1 writes to the optional debug data set specified on the System Parameters panel (ADB2Z2Z). In the example shown in Figure 9-2, the Optional Debug data set name is **LUCYK.ADBDIAG1.MSG**.

```
ADB2Z2Z n ------------------- DSN8 System Parameters -------------------- 14:1
Option ===>

    1  - Display Parameters/Generate DSNZPARM source     DB2 System: DSN8
    2  - Assemble and Linkedit DSNZPARM module           DB2 SQL ID: LUCYK
    3A - SET SYSPARM LOAD(          )
    3B - SET SYSPARM RELOAD
    3C - SET SYSPARM STARTUP
                                                                More:     +
Output datasets:
  DSNZPARM Source  ===> LUCYK.DB2TOOLS.JCL(DSN8)
  LinkEdit SYSLMOD ===> LUCYK.DB2TOOLS.LOAD(DSN8)

  Assembly listing ===> LUCYK.ASM.LIST
  LinkEdit listing ===> LUCYK.LKED.LIST
  Optional Debug   ===> LUCYK.ADBDIAG1.MSG

Input datasets:
  Assembly STEPLIB ===>
  Assembly SYSLIB  ===> 'DSN710.SDSNMACS'
                   ===> 'SYS1.MACLIB'
```

Figure 9-2 System Parameters (ADB2Z2Z)

Enter **debug** on the command line of the system parameters panel and you are prompted to enter Debug options. See Figure 9-3.

```
DB2 Admin ------------------------- Debug Options ------------------------ 13:51
Option ===>

Change one or more debug options below:

  General debug (DEBUG)   ===> DEBUG    (DEBUG or blank)
    Level       (DEBUGL)  ===> 2        (0-9)

  Panel level   (DEBUGP)  ===> 2        (0-9)
    Limit to panels  ===> ALL

  Exec level    (DEBUGE)  ===> ALL      (Off, All, Results, Commands, ...)

  VPUT to shared pool     ===> YES      (Yes or No)
```

Figure 9-3 Debug Options (ADBDEBUG)

Archived

Once the Debug options are entered, you should PF3 back to the System Parameters panel and enter **1** on the command line and at the bottom of the screen you can see the message, `Complete diagnostic data has been captured to the diagnostic file`. See Figure 9-4.

```
ADB2Z2Z n ------------------- DSN8 System Parameters -------------------- 13:54
Option ===>

    1  - Display Parameters/Generate DSNZPARM source     DB2 System: DSN8
    2  - Assemble and Linkedit DSNZPARM module           DB2 SQL ID: LUCYK
    3A - SET SYSPARM LOAD(          )
    3B - SET SYSPARM RELOAD
    3C - SET SYSPARM STARTUP
                                                                 More:     +
Output datasets:
  DSNZPARM Source  ===> LUCYK.DB2TOOLS.JCL(DSN8)
  LinkEdit SYSLMOD ===> LUCYK.DB2TOOLS.LOAD(DSN8)

  Assembly listing ===> LUCYK.ASM.LIST
  LinkEdit listing ===> LUCYK.LKED.LIST
  Optional Debug   ===> LUCYK.ADBDIAG1.MSG

Input datasets:
  Assembly STEPLIB ===>
_____________________________________________________________________
| Complete diagnostic data has been captured to the diagnostic file. |
_____________________________________________________________________
```

Figure 9-4 System Parameters (ADB2Z2Z)

Csect ADB27SPC allocates this VIO data set via:

```
ALLOC F(ADBDIAG) NEW REUSE RECFM(F) LRECL(132) SPACE(5 50) + TRACKS DSORG(PS)
```

We recommend that you allocate a similar data set manually, prior to running the function. This newly allocated data set can be used to save the data from ADBDIAG. To see if an ADBDIAG VIO data set is already allocated, use TSO ISRDDN on the command line while in DB2 Admin.

Archived

See Figure 9-5 for an example of the ADBDIAG VIO data set.

```
                          Current Data Set Allocations            Row 1 of 74

 Volume   Disposition Act DDname   Data Set Name   Actions: B E V M F C I Q
 *VIO     NEW,DEL    >    ADBDIAG  SYS07065.T170413.RA000.LUCYK.R0144346
 RESI4B   SHR,KEEP   >    ADMPC    SYS1.SADMPCF
 REST42   SHR,KEEP   >    ADMSYMBL SYS1.GDDM.MASTER.ADMSYMBL
 *VIO     NEW,DEL    >    ISPCTL0  SYS07065.T082338.RA000.LUCYK.R0128048
 *VIO     NEW,DEL    >    ISPCTL1  SYS07065.T082338.RA000.LUCYK.R0128049
 *VIO     NEW,DEL    >    ISPCTL2  SYS07065.T082338.RA000.LUCYK.R0128050
 *VIO     NEW,DEL    >    ISPLST1  SYS07065.T082338.RA000.LUCYK.R0128053
 *VIO     NEW,DEL    >    ISPLST2  SYS07065.T082338.RA000.LUCYK.R0128054
 REST41   SHR,KEEP   >    ISPMLIB  STL.LOCAL.ISPMLIB
 RESI4B   SHR,KEEP   >             SPF.PRODUCT.ISPMLIB
 MVSPP1   SHR,KEEP   >             SPF.PP.ISPMLIB
 SM4108   SHR,KEEP   >    ISPPLIB  CMERCAD.SETUP
 SM4103   SHR,KEEP   >             PTLI.DAMI310.L2C.ISPPLIB
 REST41   SHR,KEEP   >             STL.LOCAL.ISPPLIB
 RESI4B   SHR,KEEP   >             SPF.PRODUCT.ISPPLIB
 MVSPP1   SHR,KEEP   >             SPF.PP.ISPPLIB
Command ===>                                              Scroll ===> CSR
 F1=Help    F2=Split   F3=Exit    F5=Rfind   F7=Up      F8=Down    F9=Swap
```

Figure 9-5 Current Data Set Allocations using TSO ISRDDN

If you do not find that the ADBDIAG has been already allocated, then you can allocate a new ADBDIAG by issuing the appropriate command on a panel of DB2 Admin:

```
panel ISPCMDE
```

This is shown in Figure 9-6.

```
ADB21S in ----------------- DB8A Table Spaces ------------- Row 1 to 31 of 57
Command ===> panel ispcmde                                        ===> CSR

Commands: GRANT  MIG  DIS  STA  STO
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line commands

Select Name     DB Name   Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
       *        *             * *      * * * * *      *           *      * * *
------ -------- -------- ------ ------ - - - - - ------ ----------- ------ - -
*.0    ADBSCC   ADBDCC        0 BP1    R N A N Y      2          -1      0   Y
```

Figure 9-6 Allocating a new ADBDIAG

Then you see the ISPF Command Entry Panel, in which you can enter the following TSO command to allocate a new ADBDIAG data set:

```
tso ALLOC F(ADBDIAG) NEW RECFM(F B A) LRECL(133) SPACE(5 50)
DSORG(PS)  BLKSIZE(132) DSORG(PS) REUSE
```

See Figure 9-7 for the command entry on the ISPF Command Entry Panel.

```
 ISPCMDE                    ISPF Command Entry Panel

 Enter TSO commands below:
 ===> tso ALLOC F(ADBDIAG) NEW RECFM(F B A) LRECL(133) SPACE(5 50)
DSORG(PS)  BLKSIZE(132) DSORG(PS) REUSE
```

Figure 9-7 ISPF Command Entry Panel

Run the function that produces the error.

Before going back with PF3 and doing another Admin function, go to the ADBDIAG data set using TSO ISRDDN command. Browse (B) the ADBDIAG data set. Use cut (with C99999 on first line of data) and copy the ADBDIAG file into the data set you wanted to keep. Send this documentation to the Support Center along with the other documentation gathered about the problem.

9.3 Data conversion

There are two new functions to handle data conversion. They are in both V7.1.0 and V7.2.0 of DB2 Admin and Object Compare Tools.

- Stop conversion is part of Object Compare and affects how a WSL processes.
- Skip conversion is part of DB2 Admin Alter ALC function.

The Skip conversion benefits you with improved performance during ALC processing of very large amounts of data. Stop conversion protects the original object and data from being dropped.

Skip Conversion in DB2 Admin does not generate a conversion step when:

- The table space or database in which the table resides is renamed.
- The table is renamed.
- The VOLATILE attribute of the table is changed.
- Columns in the table are:
  - Moved
  - Renamed
  - Deleted
  - Inserted with an attribute of WITH DEFAULT or NULLS
- Only the attributes of a column are changed.

For further information on data conversion, see Table 9-1, extracted for reference from the *IBM DB2 Administration Tool for z/OS User's Guide and Reference Version 7 Release 2,* SC18-9552.

Archived

Table 9-1 Matrix for data type and length changes that do not require data conversion

From data type	To data type										
	SMALL INT	INT	DEC	FLOAT	CHAR	VAR CHAR	LVAR CHAR	DATE	TIME	TIME STAMP	RID
SMALL INT	Y	Y	A	Y							
INT		Y	A	Y							
DEC	A	A	A	Y							
FLOAT (1-21)				Y							
FLOAT (22-53)				Y							
CHAR	A	A	A		Y	Y	Y				
VAR CHAR	A	A	A		A	A	A				
LVAR CHAR					Y	Y	Y				
DATE								Y			
TIME									Y		
TIMESTAMP								Y	Y	Y	
RID											Y
Y = YES, data conversion is skipped unless Z is chosen as the action on DB2 Admin ALTER Column Truncation panel (ADB27CT) if you are prompted to specify what to do if the new length is smaller than the old length.											
A = ACTION, data conversion is skipped unless T or Z is chosen as the action on the DB2 Admin ALTER Column Truncation panel.											

Archived

9.3.1 Stop conversion

In the past, Object Compare would build a WSL invoking data conversion. When WSL was run and data conversion errors were found, the job would run to completion with rc=4. The job output showed:

- The object was unloaded.
- Data conversion errors were found.
- The old object was dropped.
- The new object was created.
- Data attempted to be loaded - but failed with rc=04.
- Records involved with the data conversion errors were discarded.

With the new function of Stop on conversion, the WSL process end with Rc=28 in the data conversion step, and further processing does not continue. Hence the original object and data are maintained. If you have multiple tables changes in one WSL — for example table1, table2, table3 and table4 — and all of the tables are changed, and all the changes result in a requirement for data conversion, then WSL stops on the first data conversion error found.

If the first data conversion error was found in table2, then you can see the results of data conversion in table1 and table2 but not in table3 and table4. All processing of all tables stops after the first data conversion error. To see if there are data conversion errors in table3 and table4, the WSL must be corrected and restarted.

To enable this function, you specify **YES** on a new option of the Generate Compare Jobs panel. See Figure 9-8.

```
Compare -------------------- Generate Compare Jobs  -------------------------
Option ===>

 Specify the following for DB2 Object Comparison Tool:
                                                                   More:   -
       Member prefix . . . . : APPLY    (default APPLY )
    As work statement list. : YES      (Yes/No to append to work stmt list)
    Use utility options . . : NO       (Yes/No)
    Unload Method . . . . . . U        (Unload, Parallel unload, HPU)
    Generate templates. . . : NO       (Yes/No)
    Stop on conversion error: YES      (Yes/No)

    Optional jobs after Reload or Alter:
       Run CHECK DATA  . . . : NO       (Yes/No)
       Take an image copy  . : N        (after: Reload/Alter/Both/None)
       Run REORG/REBUILD . . : A        (Mandatory, All relevant, None)
       Run RUNSTATS  . . . . : N        (after: Reload/Alter/Both/None)
       Run REBIND  . . . . . : NO       (Yes/No)

 BP - Change batch job parameters
 TU - Specify TEMPLATE usage
 UO - Change utility options
```

Figure 9-8 Generate Compare Jobs panel (GOC5)

When WSL is run, the data conversion step tests and reports all data conversion errors. If a data conversion error is found, the job fails with RC=28. The job output is shown in Figure 9-9.

```
 J E S 2  J O B  L O G  --  S Y S T E M  S Y 4 A  --  N O D E

17.15.03 J0002289 ---- THURSDAY,  15 FEB 2007 ----
17.15.03 J0002289  IRR010I  USERID LUCYK    IS ASSIGNED TO THIS JOB.
17.15.03 J0002289  ICH70001I LUCYK LAST ACCESS AT 17:12:03 ON THURSDAY, FEBRU
17.15.03 J0002289  $HASP373 LUCYWK   STARTED - WLM INIT  - SRVCLASS BATCH_M  -S
17.15.08 J0002289  IGD103I SMS ALLOCATED TO DDNAME U001
17.15.08 J0002289  IGD101I SMS ALLOCATED TO DDNAME (V001    )  106
   106                     DSN (LUCYK.K@35768.K@35768.ULC.T0001            )
   106                     STORCLAS (NORMAL) MGMTCLAS (LARGE) DATACLAS (
   106                     VOL SER NOS= SM4220
17.15.08 J0002289  ACTCC01I LUCYWK   STEP:DB2B     PROC:        COND_CODE:  28
.
.
.
ADB27AC : SERVICE LEVEL V7.1  PK34839 2006-11-17 13:15:00
-----------------------------------------------------------------------
ADB27AC - Conversions for Alter Column(s)
-----------------------------------------------------------------------

          ADBC001I Alter for table OA1A.TSN187A1
          ADBC033W Conversion of Character to Date may fail for col SN187MAT
          ADBC033W Conversion of Character to Date may fail for col SN187BPD
.
.
.
         ADB27AC - LOAD Conversion statistics

 OA1A.TSN187A1

          Column name                    Fromtype Totype   Nbr.errors
          ------------------------------ -------- -------- ----------
          SN187MAT                       CHAR     DATE        1
          SN187BRL                       DECIMAL  DECIMAL     0
          SN187BFT                       CHAR     CHAR        0
          SN187BPF                       CHAR     CHAR        0
          SN187BPD                       CHAR     DATE        1
.
.
.
          Number of Unload records selected   :         1
          Number of Unload Records skipped    :         0
          Number of Conversion errors         :         2
.
.
.
          ADBC203E Data conversion errors found
          ADBC204E Check LOAD Conversion statistics report
          ADB27AC - Ended with errors
```

Figure 9-9 WSL job output with Data conversion errors

Note: This function has been implemented via maintenance, see APARs PK34116 and PK34839 (V7.1), and PK36230 and PK36231 (V7.2).

9.3.2 Skip conversion

Db2 Admin has a new function (Skip Convert) to improve performance during the Alter (ALC) process. The Run Mode field has been replaced by the Action field, which can contain the new value of DROP-SC or DROP. DROP-SC indicates that DB2 Admin has determined that the job step to convert the data can be omitted because data conversion is not necessary.

When altering (ALC) a table, a DROP/CREATE is sometimes required. During the ALC WSL and JCL runtime, the data is unloaded and the SYSPUNCH is marked as DROP or DROP-SC (DROP Skip Convert). If a change to the data is necessary, it is marked DROP and data is converted as required for each table. When no data conversion is required it is marked DROP-SC and data conversion is avoided.

Two panels of DB2 Admin shows if the table is DROP (data conversion is done) or DROP-SC (data conversion is avoided). The Alter Table panel (ADB27C) has a new ADD primary command to enable you to add tables and the multiple tables are shown on Alter Tables panel (ADB27CA). Alter Table (ADB27C) has a new Action keyword and it shows DROP-SC or DROP once the alter process been updated.

Figure 9-10, Figure 9-11, and Figure 9-12 show some examples of Skip conversion.

```
DB2 Admin ------------------- DSN8 ALTER Table -------------- Row 1 to 6 of 16


 New owner ===> LUCYK    >                    Old owner : LUCYK
 New name  ===> CMTX200            >          Old name  : CMTX200
 Volatile  ===> NO        Action    : DROP-SC  New DB ===> DB05312
 Rows/Page    : 43.574    Partitions: 16       New TS ===> CMTTA01T
 Commands :            PRIMKEY            ALTPART
 Line commands : UP - Update Primary Key
  I - Insert  U - Update  D - Delete  R - Repeat  LAB - Label  COM - Comment
  M - Move    A - After   B - Before  X - Index   RES - Reset update
                                                                 Old
Operation
 Select Column Name        Col No Col Type Length  Scale Null D Col No Type
        *                     * *             *      * *    *      * *
 ------ -----------------> ------ -------- ------ ------ ---- - ------ --------

 *      CMT20022               1 DECIMAL       2      0 N    N      1 UPDATE
        CMT20002               2 DECIMAL       1      0 N    N      2
        CMT20003               3 DECIMAL       8      0 N    N      3
        CMT20004               4 DECIMAL       2      0 N    N      4
        CMT20005               5 DECIMAL       2      0 N    Y      5
        CMT20006               6 CHAR         30      0 N    N      6
```

Figure 9-10 ALTER Table (ADB27C)

The example is an alter to the column (rename) CMT20001 to CMT20022 which causes *no* data conversion hence - Action: DROP-SC.

If you do an ADD from the Alter Table panel, the new Alter Tables panel (ADB27CA) is presented and the field Object Action shows DROP-SC. See Figure 9-11.

```
DB2 Admin ----------------------- DSN8 Alter Tables ---------- Row 1 to 1 of 1

Commands:
  ALTER - generate jobs   ADDFK - Add Foreign Key-affected tables
Line commands:
  A - Alter Object  D - Delete  S - Select Object  REL - Alter related
 FK - Add Foreign Key-affected tables  E - Edit DDL  CD - Convert to Drop
 RS - Reset RI-FK flags

To add a table, provide an owner and name below and hit Enter
   Owner   ===>          >            (Creator)
   Name    ===>                    >  (Table Name.  ? to look up)
                                                   RI RI  FK  Chg Object
Sel Creator  Name               T DB Name  TS Name    Rels Add Add Rqd Action
    *        *                  * *        *           * *   *    *  *
--- -------- ------------------ - -------- -------- ------ --- --- --- -------
*   LUCYK    CMTX200            T DB05312  CMTTA01T      0 NA  YES     DROP-SC
****************************** END OF DB2 DATA ******************************
```

Figure 9-11 Alter Tables (ADB27CA)

Archived

The JCL is shown in Figure 9-12.

```
LUCYK.REDBK1.BATCH(CUSTALC1)
.
.
.
//*********************************************************ADB27ACM*
//* STEP DELCV40:    DELETE OLD CONVERTED FILES
//*
//*                  RESTART POINT IF PREVIOUS RUN ABNORMAL TERMINATED
//*                  IN THE FOLLOWING CONVERT STEP CNVRT50
//*
//* RESTART JOB STEP NAME IF FAILED IN THIS STEP : DELCV40
//*
//*******************************************************************
//DELCV40  EXEC PGM=IEFBR14
//DDLOUT   DD DSN=LUCYK.REDBK1.REDBOOK1.DDL.CONV,
//            SPACE=(TRK,(0,1)),
//            UNIT=SYSDA,DISP=(MOD,DELETE,DELETE)
//* OBJACTION = DROP-SC
.
.
.
//******************************************************************
//* STEP CNVRT50: CONVERT DDL, LOAD CONTROL AND DATA FOR ALTER
//*
//* RESTART JOB STEP NAME IF FAILED IN THIS STEP : DELCV40
//*
//* INTFIN   DDNUM  TABLE                                     ACTION
//* -------- -----  ---------------------------------------- ------
//* T1       001    LUCYK.CMTX200                             DROP-SC
//*
//*******************************************************************
.
.
.
//*********************************************************ADB27TL*
//* STEP LOAD001: LOAD TABLE : LUCYK.CMTX200
//*
//* RESTART STEP NAME IF FAILED IN THIS STEP : LOAD001.DSNUPROC
//*
//* TO RESTART, SPECIFY THE STEPNAME IN THE JOB CARD AND
//* UNCOMMENT THE FOLLOWING JCL STATEMENT BELOW:
//* //*           UTPROC='RESTART(PHASE)',
//* TABLE SPACE PARTITIONS = 16
//* OBJACTIO = DROP-SC
//*******************************************************************
```

Figure 9-12 JCL example of DROP-SC

Archived

An example of DROP (when data conversion *does* take place) is shown in Figure 9-13, Figure 9-14, and Figure 9-15.

Figure 9-13 shows an alter to the column CMT20002 from DECIMAL 1 to CHAR 10, which causes data conversion, hence Action: DROP.

```
DB2 Admin ------------------- DSN8 ALTER Table -------------- Row 1 to 6 of 16

New owner ===> LUCYK    >                      Old owner : LUCYK
New name  ===> CMTX200            >            Old name  : CMTX200
Volatile  ===> NO        Action    : DROP      New DB ===> DB05312
Rows/Page    : 39.766    Partitions: 16        New TS ===> CMTTA01T
Commands :            PRIMKEY            ALTPART
Line commands : UP - Update Primary Key
 I - Insert  U - Update  D - Delete  R - Repeat  LAB - Label  COM - Comment
 M - Move    A - After   B - Before  X - Index   RES - Reset update
                                                                Old Operation
Select Column Name        Col No Col Type Length  Scale Null D Col No Type
       *                       * *            *      * *    *     * *
------ -----------------> ------ -------- ------ ------ ---- - ------ ---------

       CMT20022                1 DECIMAL       2      0 N    N      1
*      CMT20002                2 CHAR         10      0 N    N      2 UPDATE
       CMT20003                3 DECIMAL       8      0 N    N      3
       CMT20004                4 DECIMAL       2      0 N    N      4
       CMT20005                5 DECIMAL       2      0 N    Y      5
       CMT20006                6 CHAR         30      0 N    N      6
```

Figure 9-13 Alter Table (ADB27C)

If you select **ADD** from the Alter Tables panel (ADB27C) then you see the Alter Tables panel (ADB27CA) with Object Action = DROP. See Figure 9-14.

```
DB2 Admin---------------------- DSN8 Alter Tables ---------- Row 1 to 1 of 1

Commands:
  ALTER - generate jobs   ADDFK - Add Foreign Key-affected tables
Line commands:
  A - Alter Object  D - Delete  S - Select Object  REL - Alter related
 FK - Add Foreign Key-affected tables  E - Edit DDL  CD - Convert to Drop
 RS - Reset RI-FK flags

To add a table, provide an owner and name below and hit Enter
   Owner   ===>          >            (Creator)
   Name    ===>                    >  (Table Name.  ? to look up)
                                                   RI RI  FK  Chg Object
Sel Creator  Name               T DB Name  TS Name    Rels Add Add Rqd Action
    *        *                  * *        *            * *   *    *  *
--- -------- ------------------ - -------- -------- ------ --- --- --- -------
*   LUCYK    CMTX200            T DB05312  CMTTA01T      0 NA  NA      DROP
****************************** END OF DB2 DATA ******************************
```

Figure 9-14 Alter Tables (ADB27CA)

The JCL is shown in Figure 9-15.

```
LUCYK.REDBK2.BATCH(CUSTALC1)
 //*************************************************************ADB27ACM*
 //* STEP DELCV40:    DELETE OLD CONVERTED FILES
 //*
 //*                  RESTART POINT IF PREVIOUS RUN ABNORMAL TERMINATED
 //*                  IN THE FOLLOWING CONVERT STEP CNVRT50
 //*
 //* RESTART JOB STEP NAME IF FAILED IN THIS STEP : DELCV40
 //*
 //*********************************************************************
 //DELCV40  EXEC PGM=IEFBR14
 //DDLOUT   DD DSN=LUCYK.REDBK2.REDBOOK1.DDL.CONV,
 //            SPACE=(TRK,(0,1)),
 //            UNIT=SYSDA,DISP=(MOD,DELETE,DELETE)
 //* OBJACTION = DROP
.
.
.
//********************************************************************
//* STEP CNVRT50: CONVERT DDL, LOAD CONTROL AND DATA FOR ALTER
//*
//* RESTART JOB STEP NAME IF FAILED IN THIS STEP : DELCV40
//*
//* INTFIN   DDNUM  TABLE                                      ACTION
//* -------- -----  ------------------------------------------ ------
//* T1       001    LUCYK.CMTX200                               DROP
//*
//********************************************************************
.
.
.
//*************************************************************ADB27TL**
//* STEP LOAD001: LOAD TABLE : LUCYK.CMTX200
//*
//* RESTART STEP NAME IF FAILED IN THIS STEP : LOAD001.DSNUPROC
//*
//* TO RESTART, SPECIFY THE STEPNAME IN THE JOB CARD AND
//* UNCOMMENT THE FOLLOWING JCL STATEMENT BELOW:
//* //*           UTPROC='RESTART(PHASE)',
//* TABLE SPACE PARTITIONS = 16
//* OBJACTIO = DROP
//*********************************************************************
```

Figure 9-15 JCL example of DROP

9.4 Limitations with CM

The following limitations are present with Change Management:

- Revoke statements can only be run as a fast change.
- Grant on plan, package, and system privileges can only run as a fast change.
- Recover does not generate revoke statements.
- Fast changes are run immediately upon registration, hence you cannot analyze or run them manually. You also cannot modify fast changes, recover them, or promote them to other systems.
- Modifying an existing change is considered a manual intervention and is not recommended.

For details, refer to *IBM DB2 Administration Tool for z/OS User's Guide and Reference Version 7 Release 2,* SC18-9552, Chapter 20 - "Modifying a change.".

Archived

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this IBM Redbooks publication.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 383. Note that some of the documents referenced here may be available in softcopy only.

- *DB2 for z/OS Tools for Database Administration and Change Management*, SG24-6420
- *Administration Solutions for DB2 UDB for z/OS*, SG24-6685
- *IBM DB2 Data Archive Expert for z/OS: Put Your Data in Its Place*, SG24-7080
- *Disk Storage Access with DB2 for z/OS*, REDP-4187

Other publications

These publications are also relevant as further information sources:

- *IBM DB2 Administration Tool for z/OS User’s Guide and Reference Version 7 Release 2,* SC18-9552-03
- *IBM DB2 Object Comparison Tool for z/OS User’s Guide Version 7 Release 2,* SC18-9554-03
- *z/OS V1R7.0 ISPF Dialog Developer's Guide*, SC34-4821-04
- *IBM DB2 Data Archive Expert User’s Guide and Reference Version 2 Release 1*, SC18-9696-02

Online resources

These Web sites are also relevant as further information sources:

- IBM DB2 and IMS Tools:

  http://www.ibm.com/software/data/db2imstools/

How to get IBM Redbooks

You can search for, view, or download IBM Redbooks, IBM Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

ibm.com/redbooks

© Copyright IBM Corp. 2007. All rights reserved.

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Archived

Index

A
ADB 18–19, 21, 102, 104, 149, 192–193, 345, 357
ADB27SPC 33, 371
ADB2ATH 16–17
ADB2CUST 10, 19, 23–24, 30
ADB2UTIL 16–17
ADB2Z2Z 370–371
ADBBIND 18, 29, 33, 38–39, 43
ADBCHG 17, 26, 122–123, 135, 145, 164
ADBCHGS 26, 122, 164
ADBCHKPT 89–90
ADBCVERSCOPE 26, 122, 188
ADBDIAG 104, 149, 357, 368–369, 371
ADBDIAG VIO 371–372
ADBL 18, 20–22, 27, 29–30, 33, 39
ADBL setting 21
ADBTEP2 50–52, 88–91, 94, 104, 162, 342, 345, 352, 357–358, 369
  Batch statement list 52
  Restart 50
ADD 77, 118, 125, 159, 173, 181–184, 206, 213, 215, 228–229, 235, 264–266, 274–275, 334, 337, 350–351, 353, 362, 377–378, 380
Add
  A catalog index 21
add DB2 Compare 27
ADDFK 278–279, 281, 378, 380
Additional indexes 366
ADDRI 215, 230–231
AL 95, 110, 113–114, 118–119, 126, 133, 163, 190, 257–258, 260–261, 263–264, 267–268, 282, 367
ALC 10, 113–114, 118–119, 125–126, 135, 163, 172–173, 176, 196–198, 257–258, 260–261, 263, 273, 282–283, 312, 334–336, 347, 355, 358–359, 367, 369, 373, 377
ALTER 5, 51, 77, 91, 99–100, 103, 106, 114, 119, 122, 126, 131–132, 137, 159, 165, 173, 177–178, 180, 196, 206, 260, 264–265, 267, 274–277, 282, 317, 325–326, 334, 336–337, 344, 348, 362–363, 374, 377–381
ALTER ALC 197
ALTER TABLE 77, 119, 122, 159, 180–184, 265–267, 328, 336–337, 344, 348, 362
AN 42, 118, 125, 130, 247, 252–253, 268, 272–273, 316, 326, 328, 336–337, 339
ANALYZE 114, 121, 126, 198, 227, 248, 270, 283–284, 301, 327–328, 331, 336, 338
ANALYZE process 139, 141, 143, 362
analyze process 20, 158–159, 186, 355
AND 22, 40, 90, 104, 145, 149, 155, 235, 352, 379, 381
AO 53, 55, 67, 80–81, 214, 262
APPLY 139, 179–180, 244, 298, 322–323, 375
AT 95, 133, 342, 350, 356, 376
AUTH 20, 24
AUTHID 108, 167, 234, 241, 256
authorization override 106
AUTHPGM 16
AUTHTSF 16

B
BALANCE 290–291
Base Version 191–193, 258
Batch
  Batch statement list 52
  Parameters 49–50
bl2llib 29
BPNAME 108, 167
BR 63, 132
BROWSE 43, 62–63, 119, 317, 351
browse xxviii, 63, 70, 110, 119

C
CAN 52, 75, 85, 130, 358, 361
Catalog navigation 54
Catalog statistics 217, 224, 233
CHANGE 253
Change
  PRIQTY 17
Change Management objects 17–18
Change Management tables 18, 25–26, 38–39, 169, 195
Change reporting options 141, 147, 150, 160, 244, 254, 270–272, 284, 286–287, 298, 323, 338, 341, 348
Change Status 162
  ANALYZED 162
  CANCELED 162
  COMPLETE 162
  DEFINED 162
  FAILED 162
  INITIAL 162
  RUNNING 162
change terminology 138, 172
Change Type
  COMPARE 154, 163, 315, 360
  FAST 163
  NORMAL 163
  PROMOTE 157, 163, 251, 255, 327, 361
  RECOVER 163, 285, 338, 349
Change Types 163
CHECK DATA 141, 160, 217–218, 245, 270, 284, 298, 323, 375
Checkpoint table 30, 88
CLASS 50, 102, 104, 149, 192, 352, 356
Classifying changes performed using DB2 Admin 260
CLIST
  ADBL 20
Clone 94–95, 104–105, 108–110, 144, 203–204, 340
Cloning 95, 105, 203–204, 222, 227, 248, 369
  A Work Statement List 105
CLOSE

© Copyright IBM Corp. 2007. All rights reserved.

NO 292
YES 97, 281, 292, 326, 356
CM Level 38
cmown 24
co-existence 29
Collection name 29
COLLNAME 108, 167
Command
/ 63
ADD 213
ADDRI 215
DCL 65
DS 78, 210
GEN 204
LIKE OFF 55
LIKE ON 55
LISTC 82
PANEL 94
REFRESH 62
VALUES 291
xC 65–66
xSEL 65
Compare
Masks 240–241, 306, 321
Source 195, 237, 295, 319–320
Source (new) 237
Target 240, 296
Target (old) 240
Connect to a remote DB2 system 88
CONTINUE 138, 154, 157, 173–174, 177–178, 180, 245–246, 251, 255, 267, 274–275, 278–280, 282, 315, 324, 327, 334–335, 338–339, 344, 349, 360–362
CP 130, 356
CRE 204
Create a WSL 92
CREATE VIEW 77, 206, 336–337, 348, 363
Create/drop/label/comment on objects 69
CS 18, 33, 39, 120, 130, 137–138
Customization
DB2 Admin 19, 48
customization EXEC 19

D

data conversion 373–374, 376
DB2 Admin
Allocate a print data set 48
Batch parameters 49
BP 49
Catalog navigation 54
Catalog search 52
Customization 20, 23–24, 48
DDL 77, 201, 204, 292, 368
First time 46
Main menu 47
Parameters 20, 24, 48, 223–224, 233
PR 49, 51
WSL 5, 51, 91, 119, 144, 217, 340
DB2 Admin Main Menu 23, 26–27, 29, 116, 318
DB2 Admin overview 4
DB2 BIND 18
DB2 BINDs 29, 33, 35
DB2 catalog 4–8, 10, 28, 30, 45, 47, 54–56, 59, 64–65, 68, 72, 75, 77, 79, 93, 105, 138, 140, 142, 181, 186, 195, 206, 237–238, 240, 261, 294–297, 301–302, 318–322, 331
DB2 Object Compare overview 7
DB2 Object Compare panels 294
DB2 object links 54
DB2 system catalog copy 79
DB2 utilities 4–5, 17
DBNAME 58–59, 63–65, 68, 108, 167, 209, 234, 241, 256, 302
DBRMNAME 108, 167
DCL 65, 91, 96, 99–100
DD 23, 26, 28, 47, 99, 102, 104, 116, 149, 181, 192–193, 261, 294, 318, 325, 339, 345, 357–358, 369, 379, 381
DDL 7–8, 10, 17–18, 38–39, 42, 65, 73–75, 79, 91, 93, 95–97, 99–100, 113, 120, 130, 141, 143, 145, 152, 155–156, 163, 166, 177, 186, 196, 201, 204, 206–207, 210–212, 257, 281, 292–295, 297, 317, 319, 325, 327, 331, 335, 346, 362–363, 368, 378–381
DDL file 8, 163, 249, 292, 294–296
debug 370
default plan name 21
Default unit name 51
Delta Version 154, 157, 174, 258, 267, 282, 285, 315, 324, 327, 360
delta version file 142, 197–198, 359
Disabling Change Management 27
Display
Checkpoint table 89
Structure - DS 78, 210
Drop impact 85
drop impact 79, 85–86
DS 53, 55, 67, 78, 80–81, 99, 210, 214, 262, 357
DSN 5, 31, 43, 51, 79, 81–82, 91–92, 94, 96, 98–99, 143, 146, 149, 192–193, 208, 211, 236, 241, 316, 321, 325, 339, 345, 350–351, 357–358, 376, 379, 381
DSNDTX03 366

E

EA 53, 130, 145
EDIT 16, 21–22, 24–25, 27–28, 39–40, 52, 65, 70, 74, 149, 167, 183, 192, 219, 226, 234, 247, 325, 339, 342, 357
Enabling Change Management 23
ENFORCE NO 236
EQUAL search 55
ER 20, 130, 148–149, 154, 353
Execute SQL statements 23, 26, 28, 47, 69, 116, 181, 261, 294, 318
Executing a Work Statement List 104
EXPLAIN 5, 18, 33, 39, 43

F

fast changes 126
Fields to ignore 242
first change 172, 174, 330
FK 95, 110, 281, 378, 380

Archived

G
GBPNAME 108, 167
GEN 62, 76, 93, 95, 142, 151, 156, 204–207, 209, 211–213, 249, 259, 368
  Masking 204
Generate a WSL 114
GR 68, 83, 346
GRANT 5, 18, 26, 42–43, 51–52, 56–58, 60–63, 66–67, 73, 76–77, 83, 86, 90–91, 172, 178, 205–206, 210, 215, 229, 255, 263, 273, 288, 372
GRANTEE 108, 167
GRPNAME 108, 167

I
I 28, 47, 116, 130, 209, 258, 314, 333, 372
IBM FTP server 368
ICS 56–58, 60–62, 66–67, 76, 86, 205, 210, 215, 229, 255, 288, 372
IG 118, 125, 130, 247, 252–253, 268, 272–273, 316, 326, 328, 336–337, 339
Ignore fields 242, 301–302
ignores 41, 117, 128, 130, 175, 188, 301, 303–304, 306, 313
II11945 369
IKJTSOxx 16
IM 130
Image copy 56–58, 60–62, 66–67, 86, 288, 372
IMPORT 118, 121, 198, 211–212, 227, 256, 311, 317, 327
IMPORT process 121, 139, 155, 249, 255, 317
IN D/L/H 64
INITIAL status 164
INS 65, 69–70, 130, 166, 169, 186, 241–242, 259, 287, 304, 314, 333
installation tasks 15–16, 30
ISPDPTRC 369
ISPF table 19, 23–24, 35, 60
ISPF tables 19, 23, 62, 93, 106
IXBPNAME 108, 167
IXNAME 108, 167, 302
IXSGNAME 108, 167

J
JES2
  JOBPARM 50
JOBPARM 50, 102, 104, 149, 192, 352

L
Large CM database 366
Launchpad 30, 46–47
LIBAPRE 20–21, 27, 29, 41
LIBDEF 21, 27
LIKE 53, 55, 57, 59, 67, 80–81, 129, 132, 136, 152, 175, 188–189, 214, 262, 307, 313, 357
LIKE OFF 55, 59
LIKE ON 55, 59
Line command
  question mark (?) 56
Linklist modules 29
LISTC 82
LISTCAT 82–83

M
MA 118, 125, 130, 247, 252–253, 268, 272–273, 316, 326, 328, 336–337, 339
maintenance strategy 15, 31, 35
Manage Work Statement Lists 93–94
Manipulate WSL 94
Masking 77–78, 105–106, 109, 157, 166, 168, 204, 206–208, 306
masking 8, 76, 105–106, 109, 111, 139, 156–157, 166–167, 207, 209, 213, 217, 306, 321–322, 327–328
masks 41, 106–108, 117, 128, 130, 167, 175, 188, 237, 240, 255, 294, 297, 303, 306–307, 313, 319, 321
maxerrors 51
MIG 26, 56–58, 60–63, 66–67, 73, 76, 86, 90, 172, 205, 210, 212–213, 215–216, 219, 223, 226, 263, 273, 288, 369, 372
  ADD 215, 228–231
Migration
  Scope 224
  Using Object Compare 237
Migration panels 214, 222
Modify a WSL 96

N
NAME 58–59, 63–66, 68, 104, 108, 145, 149, 166–167, 179, 190, 235, 252, 306, 342, 352, 356, 379, 381
new version 311
Numparts 289–291

O
Object Compare xxvi, 47, 111, 119, 124, 211–212, 257, 292, 294–296, 365–366, 368, 373, 375
  Generate jobs 243
  WSL 248, 306, 375
OR 65, 70, 130, 155, 264, 334, 342
ORIGINAL 159, 279–280, 289–291, 331–332, 355, 358–359
Override 106
overview of the two tools 4
OWNER 11, 18, 24, 39, 43, 83, 108, 166–167, 170–171, 209, 302–303, 306, 322
owner 10, 18, 20, 24–25, 55, 77–78, 133, 136–138, 151, 154, 157, 191, 203, 206–207, 259, 262, 264–265, 267, 315, 321–322, 324, 327, 334–335, 338, 349, 360–361, 377–378, 380

P
package owner 18
PARMLIB 16
partitioning override 106
PATHSCHEMAS 303
PENDING CHANGES 115
Pending changes 177–178, 367
Performance considerations 64

Archived

PGMNAME 108, 167
PK34116 377
PK34839 376–377
PK36230 17, 377
PK36231 377
PK37849 366
PK38750 150
PKGNAME 108, 167
PKLIST 29, 33, 39, 43
Plan name 29
PLNNAME 108, 167
PQ 118, 125, 130, 247, 252–253, 268, 272–273, 316, 326, 328, 336–337, 339
PR 48–49, 51, 130, 153, 194, 258, 271, 314, 334
Primary key 79, 81–82, 88, 211
PRINT 49, 70–72
print data set. 69
PRODADD 20–21, 27, 29, 41
Product libraries 20
promote 121, 130, 151, 163, 186, 196–198, 258, 292, 301, 312, 359–362
PROMOTE process 150–152, 292, 311–312, 314
promote process 198
PROMPT OFF 52
PROMPT ON 52
Prompt option 49, 51, 87

Q

Q - Cloning 105
Quick start customization 29

R

RC 22, 130, 148, 158, 161, 251, 253, 324, 339, 343, 352, 376
RDEF 10, 113–114, 119, 126, 163, 282, 288–289, 367, 369
RE 87–88, 131, 269
RECOVER process 158, 161, 329, 332, 338
Redbooks Web site 383
  Contact us xxviii
REFRESH 62, 264, 303
REGISTER 114, 121, 126, 197
REGISTER process 114, 134–135
related objects 10, 17, 122, 215, 252
REPAIR utility 98
RESET 65–66, 70, 238–240, 320
Restart Monitor 88
RESTART(NO) 90
RESTART(YES) 90
Result list
  REFRESH 62
  Removing lines 62
Reverse engineering 204
REVOKE 5, 45, 51, 83–85
Revoke impact 83
revoke impact 83–84
REXX 22, 40, 100
RI 79, 215, 218, 225, 230–231, 278, 281, 366, 378, 380
RN 42, 118, 125, 130–131, 247, 252–253, 268, 272, 316, 326, 328, 336–337, 339
RST 131, 137, 162, 164
RUN 89, 91, 100, 114, 121, 126, 198, 228, 235, 248, 271, 286, 301, 327–328, 330, 340–341, 379, 381
run DB2 Compare 29
Run SQLID 77–78, 141, 160, 206–207, 217–218, 224–225, 233, 244, 270, 284, 298, 323, 338, 348
RUNSTATS 30, 51, 100, 105, 141, 160, 214, 217–218, 221, 224–225, 227, 233, 245, 270, 284, 298, 323, 366, 375

S

S01DB2 247
SARG 57
SC 68, 95, 110, 131, 173, 190, 377–379
scalability 366
SCHEMA 108, 167, 306
SEARCH 57, 59
search functions 65
Select prototyping 26, 63, 73, 90, 172, 263, 273
SET CURRENT SQLID 18, 74–75, 95, 207, 218, 326
SFNAME 108, 167
SGNAME 108, 167
Showing data set information 45, 81
SHRVARS 145
single object creation 204
Sizing the objects 17
Skip 89, 373, 377
Skip conversion 373, 377
SMP/E 15–16, 31
SORT 59–61, 87–88
sort 59–62, 87
Sort a result list 59
source 8, 10, 105–106, 110, 120, 141, 151, 163, 187, 195–196, 198, 204, 213–216, 220–223, 227, 229, 285, 292, 294, 317–320, 331, 368, 370–371
Specify Mask Fields 306
SPUFI 6, 29, 47, 77–78, 124
SQ 95, 110, 133, 190
SQL ALTER 126
SQL ID table 35–38, 41
SQL Select Prototype 65, 70
SQLID 5, 18, 74–75, 77–78, 95, 101, 108, 117–118, 141, 160, 167, 172–173, 182, 184, 206–207, 217–218, 224–225, 233, 244–245, 265, 270, 274, 284, 298–299, 323–324, 326, 338, 348
SR 131, 179, 252
ssid 24
SST3CH 220
ST 118, 125, 131, 247, 252–253, 268, 272–273, 316, 326, 328, 336–337, 339
Statement execution prompt 52, 85, 87, 92
Stop conversion 373, 375
STPNAME 108, 167
SUPERSEDE 157, 180, 344–345, 355
Suppress
  DROP of columns 244, 298, 323
  DROP of objects 244, 296–298, 323
Switch catalog copy 80
SYS1.SISPMENU 20

System catalog panel 52

T

T 26, 53, 55–56, 161, 172, 190, 205, 258, 260, 314, 322, 333–334, 337, 372
T02DB2 247
T03COMP 247
target 5, 7–8, 10, 31, 34, 106, 121, 124, 129, 136, 141–142, 151–152, 156, 163, 168, 175, 187, 195–198, 212–214, 216–218, 220–223, 285, 292–294, 296, 317–322, 331, 368
TBNAME 107–108, 167, 302, 308, 366
TGNAME 108, 167
tools overview 3
translation mask 306
TRIGGER 77, 118, 125, 206, 292
troubleshooting 365, 368
truncation warning 25
TSBPNAME 108, 167
TSNAME 63–64, 108, 167, 209, 302
TSSGNAME 108, 167
TT 56

U

U 89, 118, 125, 130–131, 137, 160–161, 165, 189, 191, 217–218, 224–225, 258–259, 268, 314, 316, 323, 333–334, 336, 375, 377, 380
UDFNAME 108, 167
UDTNAME 108, 167
Unload data 220, 235
Unload Method 218, 225, 244, 298, 323, 375
UP 102, 265, 274–276, 377, 380
Use masking 78, 207, 217, 224, 233
User-defined function 190

V

V 26, 53, 131, 144, 203, 262, 319, 340, 372
Validate 95
VALUES 155, 290–291
VCATNAME 108, 167, 302
VE 118, 125, 130–131, 247, 252–253, 259, 268, 272, 314, 316, 333, 336–337
Version xxv, 10, 16–17, 19, 21, 25, 28–29, 45–46, 79–80, 88, 91, 112, 121–122, 136, 138, 146–147, 150–151, 153–154, 186–189, 191, 197, 201, 203, 217, 238, 240, 245–246, 251, 254–255, 258–260, 313–316, 320, 324, 327, 330, 333–335, 338, 341, 349, 360–361, 367, 373
version scope 124, 142, 150, 153, 186–188, 254, 258–259, 287, 313–314, 333
version scopes 131, 188–189, 313
versions 17, 21, 29, 79, 112, 114, 117, 121, 128, 131, 139, 141, 152, 186, 188, 198–199, 258, 303, 306, 312–314, 330–331, 333, 341, 360
VIRTUAL CHANGES 115
Virtual changes 367
VSAM cluster information 82

W

Work statement list 51–52, 75, 85, 351
  Basics 91
  Name 91
Work with a WSL 93
Worklist name 88, 91, 216, 223, 233, 244, 297–298, 323
WSL 10, 17, 42, 51–52, 75, 90–92, 114, 118, 127, 131, 141, 143–144, 158, 160, 162, 197–198, 204, 207, 210–213, 217, 219–221, 224, 248, 251, 270, 284, 323, 329, 331–332, 338, 340, 345, 369, 373, 375–377
  Cloning 95, 248, 251
  Creation 92, 96
  Execution 92, 104
  ISPF table 106
  Modification 96
  Object Compare 248
  Operations on 92
  Override 106
  SELECT 42, 96
  STATEMENT 97
  TYPE 96

X

xC 53, 55, 65–68, 80–81, 214, 262
xSEL 65
xxxxxS1 221, 227
xxxxxSE 220–221, 227

Z

Z - DB2 system administration functions 88
ZONECOPY 31

Archived

Archived

DB2 for z/OS Administration Tools for Enhanced Change Management

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

Archived

Archived

Archived

DB2 for z/OS Administration Tools for Enhanced Change Management

DBA activities with IBM DB2 Object Comparison and Administration Tool

Change management functions with Version 7.2 of the two tools

Scenarios and examples

A database administrator (DBA) has a challenging job trying to maintain databases in today's nonstop 24x7 environments. This challenge is compounded by the complexity of database structures and the time pressure of trying to be up again as soon as possible without making mistakes.

When business expands, several DB2 subsystems have to be maintained, such as multiple development systems, function test, acceptance test, volume test, and production systems. Often changes have to be implemented across multiple DB2 subsystems.

The two IBM DB2 for z/OS tools that help database administrators to implement database changes are the DB2 Administration Tool and the DB2 Object Comparison Tool.

In this IBM Redbooks publication we show how, with Version 7.2, the two tools offer an enhanced Change Management function to manage and track the changes of your DB2 objects. We show how the tools help you define changes, resolve conflicts, register the changes, analyze the changes to generate a work statement list that applies the changes, and then run the changes in the correct order. The Change Management functions also allow you to back out completed changes and facilitate the management of a convenient audit trail.

This book contains a major update to the Change Management functions described in DB2 for z/OS Tools for Database Administration and Change Management, SG24-6420-01.

SG24-7441-00 ISBN 0738489360

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks